# Table of Contents

Lab Overview - HOL-1811-05-SDC - vSphere Automation with PowerCLI .................................................. 2  
Lab Guidance ............................................................................................................................................. 3  
Module 1 - Automate configuration activities for vCenter Server (30 minutes) .................................. 9  
  Introduction ............................................................................................................................................... 10  
  Installing and Upgrading VMware PowerCLI Considerations ......................................................... 12  
  VMware PowerCLI Help ...................................................................................................................... 17  
  Installing VMware PowerCLI Modules ................................................................................................. 20  
  PowerCLI Syntax, Pipelines, Wildcards, Variables, and Parameters .................................................. 23  
  Getting Started With PowerShell and PowerCLI .................................................................................. 28  
  Connect to the vCenter Server Using VMware PowerCLI ................................................................. 33  
  Create a Datacenter in vCenter Using VMware PowerCLI ............................................................... 35  
  Get and Set Cluster Settings ................................................................................................................. 40  
  Manage Resource Pools with VMware PowerCLI .............................................................................. 52  
  Create and Manage Folders with VMware PowerCLI ......................................................................... 59  
  Conclusion ............................................................................................................................................ 68  
Module 2 - Automate configuration activities for ESXI Hosts (30 minutes) .................................... 72  
  Introduction ............................................................................................................................................... 73  
  Getting Started With PowerShell and PowerCLI .................................................................................. 75  
  Connect to the vCenter Server Using VMware PowerCLI ................................................................. 80  
  List ESXi Hosts Within the vCenter Server ......................................................................................... 82  
  Adding and Removing Hosts from vCenter Servers ......................................................................... 85  
  Get Datastore Information and Rescan HBAs and VMFS Datastores ............................................. 87  
  Conclusion ............................................................................................................................................ 91  
Module 3 - Automate Virtual Machine management (30 minutes) .................................................. 94  
  Introduction ............................................................................................................................................... 95  
  Getting Started With PowerShell and PowerCLI .................................................................................. 97  
  Connect to the vCenter Server Using VMware PowerCLI ................................................................. 102  
  Managing Virtual Machines ................................................................................................................. 104  
  Working With VM Templates ............................................................................................................... 107  
  Deploying VMs from a Template .......................................................................................................... 110  
  Create VMs from an OVF Template ..................................................................................................... 111  
  Increase Disk Space in a VM ................................................................................................................. 113  
  Upgrading Virtual Hardware Version of a VM .................................................................................... 115  
  Create Basic Reports ............................................................................................................................ 117  
  Managing VM Snapshots Using PowerCLI .......................................................................................... 119  
  Conclusion ............................................................................................................................................ 124  
Module 4 - PowerCLI Advanced Topics ..................................................................................... 128  
  Create Basic Reports (copied) .............................................................................................................. 129  

---

HOL-1811-05-SDC
Lab Overview - HOL-1811-05-SDC - vSphere Automation with PowerCLI
Lab Guidance

Note: It will take more than 90 minutes to complete this lab. You should expect to only finish 2-3 of the modules during your time. The modules are independent of each other so you can start at the beginning of any module and proceed from there. You can use the Table of Contents to access any module of your choosing.

The Table of Contents can be accessed in the upper right-hand corner of the Lab Manual.

Hands-on with VMware PowerCLI. Gain familiarity with the tool, and then dive deeper into the functionalities available with real world examples. Both new and experienced users are sure to learn something new about automating their environments.

Lab Module List:

- **Module 1 - Automate Configuration of vCenter Server** (30 minutes) (Basic) Replace Text with Module Description.
- **Module 2 - Automate Configuration of ESXi Hosts** (30 minutes) (Basic) Replace Text with Module Description.
- **Module 3 - Automate Virtual Machine Management** (35 minutes) (Basic) Replace Text with Module Description.

Lab Captains:

- **Module 1 - 3 - Kevin Steil, Global Solutions Consultant, USA**

This lab manual can be downloaded from the Hands-on Labs Document site found here:

http://docs.hol.vmware.com

This lab may be available in other languages. To set your language preference and have a localized manual deployed with your lab, you may utilize this document to help guide you through the process:

Location of the Main Console

1. The area in the RED box contains the Main Console. The Lab Manual is on the tab to the Right of the Main Console.
2. A particular lab may have additional consoles found on separate tabs in the upper left. You will be directed to open another specific console if needed.
3. Your lab starts with 90 minutes on the timer. The lab can not be saved. All your work must be done during the lab session. But you can click the **EXTEND** to increase your time. If you are at a VMware event, you can extend your lab time twice, for up to 30 minutes. Each click gives you an additional 15 minutes. Outside of VMware events, you can extend your lab time up to 9 hours and 30 minutes. Each click gives you an additional hour.

Alternate Methods of Keyboard Data Entry

During this module, you will input text into the Main Console. Besides directly typing it in, there are two very helpful methods of entering data which make it easier to enter complex data.
Click and Drag Lab Manual Content Into Console Active Window

You can also click and drag text and Command Line Interface (CLI) commands directly from the Lab Manual into the active window in the Main Console.

Accessing the Online International Keyboard

You can also use the Online International Keyboard found in the Main Console.

1. Click on the Keyboard Icon found on the Windows Quick Launch Task Bar.
Click once in active console window

In this example, you will use the Online Keyboard to enter the "@" sign used in email addresses. The "@" sign is Shift-2 on US keyboard layouts.

1. Click once in the active console window.
2. Click on the Shift key.

Click on the @ key

1. Click on the "@ key".

Notice the @ sign entered in the active console window.
Activation Prompt or Watermark

When you first start your lab, you may notice a watermark on the desktop indicating that Windows is not activated.

One of the major benefits of virtualization is that virtual machines can be moved and run on any platform. The Hands-on Labs utilizes this benefit and we are able to run the labs out of multiple datacenters. However, these datacenters may not have identical processors, which triggers a Microsoft activation check through the Internet.

Rest assured, VMware and the Hands-on Labs are in full compliance with Microsoft licensing requirements. The lab that you are using is a self-contained pod and does not have full access to the Internet, which is required for Windows to verify the activation. Without full access to the Internet, this automated process fails and you see this watermark.

This cosmetic issue has no effect on your lab.

Look at the lower right portion of the screen
Please check to see that your lab is finished all the startup routines and is ready for you to start. If you see anything other than "Ready", please wait a few minutes. If after 5 minutes your lab has not changed to "Ready", please ask for assistance.
Module 1 - Automate configuration activities for vCenter Server (30 minutes)
Introduction

In this module, you will learn how to Automate configuration activities for the vCenter server using VMware's PowerCLI command-line tool.

This Module contains the following lessons:

- **Installing and Upgrading PowerCLI Considerations**: We go over some of the requirements to install and update VMware PowerCLI.
- **PowerCLI Help**: We will cover the Get-Help command as well as how to update the help content from Microsoft.
- **Installing Modules**: We cover how to install VMware PowerCLI Modules to add additional capabilities in VMware PowerCLI for our VMware environment.
- **PowerCLI Syntax, Pipelines, Wildcards, Variables, and Parameters**: We learn how to use proper Syntax as well as the use of Pipelines, Wildcards, Variables, and Parameters in PowerCLI.
- **Getting Started With PowerShell and PowerCLI**: We learn how to launch PowerShell, PowerCLI, and then open a few panes to make it easier to work with.
- **Connect to the vCenter Server Using VMware PowerCLI**: We learn how to connect to a vCenter server using VMware PowerCLI, this is the first step which needs to be done to start creating or modifying vCenter server objects.
- **Create a Datacenter in vCenter Using PowerCLI**: We learn how to create a datacenter using VMware PowerCLI.
- **Get and Set Cluster Settings**: We learn how get cluster advanced properties, modify them, as well as create a cluster using VMware PowerCLI.
- **Manage Resource Pools with VMware PowerCLI**: We learn how to manage Resource Pools using VMware PowerCLI.
- **Create and Manage Folders with VMware PowerCLI**: We learn how to create and manage folders using VMware PowerCLI.

Introducing VMware PowerCLI

VMware PowerCLI offers an excellent command-line alternative to the point-and-click based administration of the vSphere Web Clients. As virtualization has become mainstream and deployments begin to sprawl, the simple commands of VMware PowerCLI allows faster administration by executing tasks against large groups of objects in the virtual environment.

VMware PowerCLI follows a very logical pattern, it can be quickly adopted and understood, making it the first choice for many vSphere administrators. It also combines extensibility to allow users to build their own functions and modules to solve specific problems not addressed by out-of-box functionalities present in the vCenter Graphical User Interface (GUI) console.

The focus of this lab session will be on VMware PowerCLI.
Overview/Benefits

VMware helps enterprise IT overcome cloud management challenges with the following differentiators:

- **Achieving Fastest Time-to-Cloud Value**: extensive out-of-the-box capabilities, extensible architecture and ability to leverage existing investments, enables you to create scalable private, public and desktop cloud services in just days
- **Rapid Ecosystem Integration**: a full spectrum of extensibility options that empowers IT personnel to enable, adapt, and extend their cloud to work within their existing IT infrastructure and processes, thereby eliminating expensive service engagements while reducing risk
- **Embracing Consumerization of IT**: unifies disparate systems and processes into a highly personalized, self-service experience governed by user and organizational awareness, ensuring the exact needs of the user are met in the context of the business
- **Boosting IT Efficiency**: VMware's advanced self-service and automation capabilities lead to a significant reduction in Operational Expenditures (OpEx), while organizationally aware governance enables a multi-tenant infrastructure, improving hardware utilization and eliminating Capital Expenditures (CapEx)
- **Increasing Business Agility**: enterprise IT is empowered to quickly enable cloud services so that the business can react quickly to changing market demands and capture market share
- **Hybrid Cloud Automation**: agnostic provisioning of hardware platforms, operating systems, hypervisors, management tools, and public clouds; vRealize Automation allows customers to rapidly stand up cloud services to deliver quick time to value from your IT investments
Installing and Upgrading VMware
PowerCLI Considerations

In this entire lesson, we will discuss the considerations and associated steps of installing and updating VMware PowerCLI.

**NOTE:** In this particular lesson, we WILL NOT be performing ANY PowerShell or VMware PowerCLI commands in the Hands On Lab (HOL) lab environment that we will working in!

**Supported Operating Systems:**

You can install VMware PowerCLI on supported Windows operating systems. You can run guest cmdlets against virtual machines on which supported guest operating systems are installed.

- **PowerCLI Local Operating Systems:**
  
  For a list of operating systems on which you can install VMware PowerCLI, see the [Compatibility Matrixes for VMware PowerCLI](#) located [here](#).

- **PowerCLI Guest Operating Systems:**
  
  You can run VMware PowerCLI guest cmdlets against virtual machines with supported guest operating systems. For a list of operating systems on which you can install VMware PowerCLI, see the [Compatibility Matrixes for VMware PowerCLI](#) located [here](#).

  *Note*: Guest cmdlets are not compatible with IPv6 environments.

**Supported VMware Products:**

You can use the PowerCLI modules to manage all supported VMware products. For a list of VMware products with which VMware PowerCLI is compatible, see [VMware Product Interoperability Matrixes](#) located [here](#).

**Supported Windows PowerShell Versions:**

VMware PowerCLI is compatible with multiple versions of Windows PowerShell. For a list of PowerShell versions with which VMware PowerCLI is compatible, see [Compatibility Matrixes for VMware PowerCLI](#) located [here](#).

**Prerequisites for Installing and Running PowerCLI:**

Before installing and running VMware PowerCLI, verify that you have installed the required software on the same machine. For a list of software that you need if you
want to work with VMware PowerCLI, see **Compatibility Matrixes for VMware PowerCLI** located [here](#).

### Installing VMware PowerCLI

You can install VMware PowerCLI by running a Windows PowerShell command. You can install all official modules with a single command, or install modules individually. The VMware PowerCLI modules are available on the PowerShell Gallery Web site. When you run **Install-Module** from the Windows PowerShell prompt, the command downloads and installs the specified module. For a list of available VMware PowerCLI modules, see the PowerShell Gallery Web site located [here](#).

**Prerequisites:**

- Before installing VMware PowerCLI, see the prerequisites that we covered at the beginning of this chapter.
- Verify that you have uninstalled VMware PowerCLI 6.5 R1 or earlier from your system.
- Verify that you have registered the PowerShell Gallery as a local repository.
- Verify that your system is connected to the Internet.
- If you use Windows PowerShell 3.0 or 4.0, install the PowerShell Get and Package Management modules from the PowerShell Gallery Web site.

**Procedure:**

- Open the Windows PowerShell console.
- To install all official VMware PowerCLI modules, run the following command.

  ```powershell
  Install-Module VMware.PowerCLI -Scope CurrentUser
  ```

  **NOTE:** The modules are installed to the "$home\Documents\WindowsPowerShell\Modules" folder location.

### Allow Execution of Scripts

If you want to run scripts and load configuration files with VMware PowerCLI, you must set the execution policy of Windows PowerShell to "RemoteSigned".

For security reasons, Windows PowerShell supports an execution policy feature. It determines whether scripts are allowed to run and whether they must be digitally signed. By default, the execution policy is set to **Restricted**, which is the most secure policy. For more information about the execution policy and script digital signing in Windows PowerShell, run **Get-Help About_Signing**. You can change the execution policy by using the **Set-ExecutionPolicy** cmdlet.
NOTE: We will NOT be uninstalling or installing versions of PowerCLI in this lab. The above information on removing it was for INFORMATIONAL purposes only!

- Open the Windows PowerShell console.

Run `Set-ExecutionPolicy RemoteSigned`

### Updating PowerCLI Considerations

When we need to upgrade VMware PowerCLI to a new version, there are some steps that should be done prior to installing the new version of VMware PowerCLI. **If VMware PowerCLI was installed using an Microsoft .MSI file, we will need to uninstall the existing instance of Windows PowerShell prior to upgrading VMware PowerCLI.** It is also worth checking to ensure the “PowerCLI” folder has been removed from the following directory (`C:\Program Files (x86)\VMware\Infrastructure\`).

NOTE: We will NOT be uninstalling or installing versions of PowerCLI in this lab. The above information on removing it was for INFORMATIONAL purposes only!

- Open the Windows PowerShell console.
- Uninstall the existing version of the module.
Get-Module VMware.Module_Name | Uninstall-Module -Force

- Install the new version of the module.

Install-Module VMware.Module_Name

**Windows PowerShell - Online Update**

For the online install, start by confirming access to Windows PowerShell Gallery and being able to find the VMware PowerCLI module. This can be done by running the following command.

*NOTE: DO NOT run this command in the HOL lab environment! This is for informational purposes only.*

Update-Module -Name VMware.PowerCLI

**NuGet Access Verification**

If you have not accessed the Windows PowerShell Gallery before, or perhaps have an out of date version of NuGet, you may receive a message indicating there is a missing or out-of-date NuGet provider. NuGet is a Package Management provider. These are primarily used to install, upgrade, configure, and/or remove software in an automated fashion. To accept the installation of a proper version of NuGet, hit “Y”. We would now make use of the `Install-Module` cmdlet to make VMware PowerCLI actually available on the local system. This can be done with the following:

*NOTE: DO NOT run this command in the HOL lab environment! This is for informational purposes only.*

Install-Module -Name VMware.PowerCLI -Scope CurrentUser

You will notice we’re only installing it for the current user, we do this because it doesn’t require admin access! If you would like it available for all users of the computer, your Windows PowerShell session will have to be running as an administrator, and VMware PowerCLI will automatically be installed for all users by changing the Scope parameter to AllUsers.
**Windows PowerShell - Offline Update Without Internet Access**

The following method should be used to install VMware PowerCLI through the Windows PowerShell gallery for those systems which do not have access to the internet. You will need at least one system that has internet access and a way to move the files to the target computer. While on a system that has internet access, we will find the VMware PowerCLI module with the same command we ran above.

**NOTE: DO NOT run these commands in the HOL lab environment! This is for informational purposes only.**

```
Find-Module -Name VMware.PowerCLI
```

Then we can download the module for offline consumption with the following command and save it to a desired location on your system:

```
Save-Module -Name VMware.PowerCLI -Path C:\FolderPath\n
```

At this point, we’ll want to copy those downloaded folders and place them on the system without internet access in a location where Windows PowerShell can find them, this is the modules folder and can be confirmed by typing the below at the powershell prompt.

```
Local User: $home\Documents\WindowsPowerShell\Modules
All Users: $pshome\Modules
```
VMware PowerCLI Help

In this lesson, we will discuss how to get help with the VMware PowerCLI commands. It has a built-in help to assist with proper syntax of all the VMware PowerCLI commands.

Get-PowerCLIHelp

If you are looking for help with VMware PowerCLI, you can type the command "Get-PowerCLIHelp". It will then open up your default web browser and connect you to this (https://www.vmware.com/support/developer/PowerCLI/) web page providing your computer has an internet connection.

Update Help From Within VMware PowerCLI
NOTE: DO NOT run this command in the HOL lab environment that we are currently working in. This is meant for informational purposes only!

We recommend that every once in a while, you update the help content to ensure you have the latest information. I would especially do it after any VMware PowerCLI updates that you install. To get help updated, simple type "Update-Help" to get the latest content from Microsoft.

Help From Within the VMware PowerCLI Command-Line Tool

If you need help while working inside the VMware PowerCLI command-line tool itself, that is also available. To get help on any command, you can simple type "Get-Help Command_Name" to get assistance on the proper syntax of that specific command.

1. Type the following command in the upper pane of VMware PowerCLI.

   Get-Help Get-VM

1. Click on the "Run Script" icon in the menu to execute the command.
2. We now see that it provides us help on the command that we specifically ask for which was the `Get-VM` command in this case.

Help From Within the VMware PowerCLI Command-Line Tool (continued)

To get help on any command, you can simple type "`Get-Help Command_Name -Full`" to get detailed help on the command.

1. Type the following command in the upper pane of VMware PowerCLI.

```
Get-Help Get-VM -Full
```

1. Click on the "Run Script" icon in the menu to execute the command.
2. We now see that it provides us a more detailed help when we add the "-Full" attribute to it.
Installing VMware PowerCLI Modules

In this entire lesson we will discuss the VMware PowerCLI Modules.

In simplest terms, a module is just a grouping of functions and code around a central theme. Modules are created for applications (like Microsoft Exchange, Active Directory, VMware, and so on) to manage all aspects of different applications. They are created for print management, network adapter configurations, and more. There's always a central theme or object that modules are created to manage.

There are four different kinds of PowerShell modules:

- **Script Modules** - PSM1 files that typically contain mostly functions, but can contain any valid PowerShell code.
- **Binary Modules** - Compiled DLL files typically not created by IT pros; these are usually left up to developers.
- **Manifest Modules** - Script modules that contain a manifest.
- **Dynamic Modules** - Modules that are never written to disk and are only available in memory.

Each module serves specific purposes, but the module type you'll be creating will most likely be script modules. Script modules don't require knowledge of C# or the compilation process, and they are the most common, especially among IT pros.

VMware PowerCLI contains modules of cmdlets based on Microsoft PowerShell for automating vSphere, VMware Site Recovery Manager, vSphere Automation SDK, vCloud Director, vCloud Air, vSphere Update Manager, vRealize Operations Manager, and VMware Horizon administration. VMware PowerCLI provides a PowerShell interface to the VMware product APIs.
List Available VMware PowerCLI Modules

We can run the `Get-Module` command with some additional parameters to return a list of all the VMware PowerCLI modules that are installed on the system that can be used in VMware PowerCLI to manage our VMware environment.

**NOTE: DO NOT run this command in the HOL lab environment! This is for informational purposes only.**

```powershell
Get-Module VMware* -ListAvailable
```

We see the folder locations of where the PowerShell and PowerCLI modules are kept on a Windows operating systems.

**PowerShell:** "C:\Program Files\WindowsPowerShell\Modules"

**PowerCLI:** "C:\Program Files (x86)\Infrastructure\PowerCLI\Modules" (This location is only when VMware PowerCLI was installed using a Windows "*.MSI" file)

Importing VMware PowerCLI Modules

One of the great enhancements in VMware PowerCLI 6.5.1, means we no longer need to load the modules into the PowerShell session as we may have done in the past. So soon as the modules are loaded into the module folder, PowerShell will automatically be aware of their existence and you will find the cmdlets registered within the Windows PowerShell session. Normal Windows PowerShell behaviour means that as soon as you use the first cmdlet the VMware PowerCLI module will be loaded as needed.
NOTE: DO NOT run these commands in the HOL lab environment! This is for informational purposes only.

```
Import-Module VMware.PowerCLI
```

Please note the warning message about joining the Customer Experience Improvement Program (CEIP). We recommend enabling CEIP, as it’s greatly helps us improve our products! In order for the message to not appear, you either have to enable or disable CEIP by way of the `Set-PowerCLIConfiguration` cmdlet. The other option to load the cmdlets, simply use the cmdlets as you normally would. The modules will auto-populate as necessary!

Here’s an example of using a new Windows PowerShell session and connecting to a vCenter Server. We can see the session begins without any VMware PowerCLI modules being imported. After typing in the `Connect-VIServer` cmdlet (tab complete is fully operational as well), we can see the `VMware.VimAutomation.Core` module has been imported automatically! The other modules won’t be imported until they’re referenced.
PowerCLI Syntax, Pipelines, Wildcards, Variables, and Parameters

In this lesson, we will cover some of the core basics of using VMware PowerCLI. These are using the proper VMware PowerCLI Syntax, the use of the Pipe ( | ) operator, the use of Wildcards (*), and some Common Parameters.

Proper PowerShell/PowerCLI Syntax

Windows PowerShell and VMware PowerCLI cmdlets use a consistent verb-noun structure, where the verb represents the action and the noun represents the object to operate on. The cmdlets follow consistent naming patterns, ensuring that construction of a command is easy if you know the object that you want to work with. All command categories take parameters and arguments. A parameter starts with a hyphen and is used to control the behavior of the command. An argument is a data value consumed by the command.

A simple command has the following syntax:

Verb-Noun -parameter1 argument1 -parameter2 argument2, argument3

Proper Syntax for "Parenthesis" Usage

Another important thing to know in regards to using proper Windows PowerShell and VMware PowerCLI syntax to ensure success is to know when you do and do not have to use parenthesis ("My VM") around names. When you are using any of the commands and you are giving the name of a vCenter object, you **DO NOT** need parenthesis when there is no space between the words in the name. If you have a name that has a space in the name, you **MUST** put the full name within parenthesis.

**Examples:**

1. **Parenthesis NOT needed**

   ```
   Get-VM -Name MyVM
   ```

2. **Parenthesis MUST be used:**

   ```
   Get-VM -Name "My VM"
   ```
The Pipe ( | ) Operator

A pipeline is a series of commands separated by the "Pipe" Operator ( | ). Each command in the pipeline receives an object from the previous command, performs some operation on it, and then passes it along to the next command in the pipeline. Objects are output from the pipeline as soon as they become available. You can type a pipeline on a single line, or spread it across multiple lines. You can cycle backwards through command history using the up arrow, so it is easier to repeat pipelines if you type them on a single line.

```
Get-Cluster -Name RegionA01-COMP01 | Get-VMHost
```

This example uses the "Get-Cluster" command to connect to the RegionA01-COMP01 cluster, and then takes the output of that and uses it as the input to the "Get-VMHost" command. The end result returns a list of all ESXi hosts in the RegionA01-COMP01 cluster.
PowerShell has a number of pattern-matching operators named wildcards that you can use to substitute one or more characters in a string, or substitute the complete string. All wildcard expressions can be used with the VMware PowerCLI cmdlets. For example, you can view a list of all files with a .txt extension by running `Get-ChildrenItem *.txt`. In this case, the asterisk * operator matches any combination of characters. With wildcard patterns you can indicate character ranges as well. For example, to view all files that start with the letter "S" or "T" and have a "\.txt" extension, you can run `Get-ChildrenItem [st]*.txt`.

You can use the question mark (?) wildcard to match any single character within a sequence of characters. For example, to view all .txt files with names that consist of string and one more character at the end, run `Get-ChildrenItem string?.txt`.

In the example provided in the screen capture as well as the below command, we used the "Get-VM" command and then used the argument "-Name *.corp.local". So we searched for any virtual machine that had a Fully Qualified Domain Name (FQDN) that is on the "corp.local" domain.
Using Variables

When administrators use the same commands over and over again, it can be cumbersome to retype the same thing over and over again. This is especially the case with more advanced commands much longer that what we will see in this lab. So to help reduce the amount of commands they have to type out, they will sometimes create their own "variables" which you can think of as creating a shortcut.

The below commands and in the screen shot, we see how to create a variable to use over and over again to shorten the amount of typing. The first line of commands is where we create the variable name and what it represents. So we see we created the variable "$allvms" which equals the command "Get-VM -Name "*.corp.local" That essentially means get all VMs that have a Fully Qualified Domain Name (FQDN) that is on the "corp.local" domain.

```
$allvms = Get-VM -Name "*.corp.local"
```

Then the next line of code is what we would type in anytime in the future to reduce the amount of typing. So we use the "Get-VM" command to get all VMs and with the Fully Qualified Domain Name (FQDN) that is on the "corp.local" domain by using our newly created variable "$allvms".

```
Get-VM -Name $allvms
```
In a production environment, administrators will typically be typing in much longer commands and many lines of it as well. So in the production world, using the variable will prove to be more fruitful than the simple example of one we have shown here.

**VMware PowerCLI Common Parameters**

The Windows PowerShell engine retains a set of parameter names, referred to as common parameters. All PowerShell cmdlets, including the PowerCLI cmdlets, support them. Some of the Windows PowerShell common parameters are Verbose, Debug, ErrorAction, ErrorVariable, OutVariable, and OutBuffer. For a full list of the common parameters and more details on their usage, run Get-Help about_CommonParameters.


- **WhatIf** - Displays the effects of a command without running it.
- **Confirm** - Prompts for confirmation before running a command that stops a program or service, or deletes data.
Getting Started With PowerShell and PowerCLI

In this lesson, we will learn the basics of Windows PowerShell and VMware PowerCLI. We'll learn how to start the VMware PowerCLI console and Windows PowerShell ISE, how to list available commands and view their help.

Starting VMware PowerCLI

VMware PowerCLI has been pre-installed in the lab already for us.

1. **Double click** the desktop icon labeled **VMware PowerCLI**.

   **NOTE**: This will open the Windows PowerShell console and load all VMware PowerCLI modules.

Using Windows PowerShell ISE

As an alternative to the simple console, you can use the Windows PowerShell ISE script editor, which provides a better user experience.

1. **Type in** "ise" and hit the **enter key** at the PowerCLI prompt to launch ISE script editor.
Using Windows PowerShell ISE

The ISE script editor will open in the default view that looks like the screen capture.

Show Script Pane Top

1. Click on the "Show Script Pane Top" icon in the menu.
Show Commands Add-On

1. Click on the "Show Commands Add-On" icon in the menu.

VMware PowerCLI - Panes Overview

1. The **upper pane** is for viewing/editing the commands that we want to execute.
2. The **lower pane** is for running individual commands and displaying their output (an analog of the standard VMware PowerShell console). We will only use it to view the output of the commands we will enter into the upper pane.
3. To the **right hand side** is a **Commands** window that lets you lookup commands by there respective module.
Executing Get-VICommand Command in VMware PowerCLI ISE

During this lab, we will executing VMware PowerCLI's modules using Windows PowerShell ISE. You can type the commands in the ISE console. Each command can be selected and executed by pressing F8 (or "Run Selection" button). Before executing a new command make sure the previous one is completed, you will see the "Completed" message at the bottom of the console.

VMware PowerCLI's Modules

Let's explore what we can do with VMware PowerCLI. VMware PowerCLI's snapin's provide more than 600 commands (called cmdlets in Windows PowerShell) for managing vSphere, Site Recovery Manager (SRM), vRealize Operations Manager (vROps), and vSphere Update Manager (VUM). You can view the available VMware PowerCLI commands by typing "Get-VICommand".
This will list all VMware PowerCLI commands. As the list is quite large, you may want to narrow it down to something more specific, for example all commands for managing VMs.

1. Type the following command into the top text pane.

```
Get-VICommand *VM
```

2. Click on the **Run Script** button in the menu to execute the command,
3. You will see the numerous cmdlets listed in the **Output Pane**

**NOTE:** You can use auto complete for faster typing. Just start typing the beginning of the command/parameter, and press the "Tab" key until the specific command you want is listed.
Connect to the vCenter Server Using VMware PowerCLI

This lesson helps define the steps which needs to be performed to connect to a vCenter server. To execute any cmdlet in VMware PowerCLI, you need to connect to the vCenter using the `Connect-VIServer` cmdlet.

By default, VMware PowerCLI cmdlets run on the vCenter Server systems or vCloud Director (vCD) servers you are already connected to or whichever is the default. When you connect to a vCenter Server system by using the `Connect-VIServer` cmdlet, the server connection is stored in the `$DefaultVIServers` array variable. This variable contains all connected servers for the current VMware PowerCLI session. To remove a server from the `$DefaultVIServers` variable, you can either use `Disconnect-VIServer` to close all active connections to this server, or modify the value of `$DefaultVIServers` manually.

When you connect to a vCloud Director system by using `Connect-CIServer`, the server connection is stored in the `$DefaultCIServers` array variable. This variable contains all connected servers for the current session. To remove a server from the `$DefaultCIServers` variable, you can use `Disconnect-CIServer` cmdlet.

Connect to the vCenter Server

The first step is to connect to our vCenter server. You need to use the `Connect-VIServer` cmdlet. Also note if you are following the previous module you can ignore this step as you are already connected to the vCenter server.
1. Type in the following command into the upper text pane.

   Connect-VIServer vcsa-01a.corp.local -User administrator@corp.local -Password VMware1!

2. Click on the "Run Script" icon to execute the command.

3. When you have successfully logged into the vCenter server, a prompt will be displayed that shows the server you are connected to, using Port 443, and the user account in the output pane.
Create a Datacenter in vCenter Using VMware PowerCLI

In this lesson we will explore steps to create a datacenter object in vCenter.

vSphere objects are used to create virtual datacenters. The Datacenter is at the root of vSphere and allows the clusters and other host infrastructures to be placed inside the virtual datacenter. Most VMware administrators use datacenter objects in vSphere to represent a physical site and datacenter objects are used to define the boundary where the infrastructure exists.

Search for Existing Datacenters

VMware PowerCLI has cmdlets which allow you to search for existing datacenters. To start, let’s run the `Get-Datacenter` cmdlet with no additional parameters to see whether there are any existing objects in vCenter.

1. Type in the following command in the upper text pane.

   ```
   Get-Datacenter
   ```

2. Click on the "Run Script" icon to execute the command.
3. You will see that the command returns the name of an existing datacenter called "RegionA01" datacenter in the lower pane.
Create a New Datacenter

Now let's create a new datacenter using VMware PowerCLI and call it "MyDatacenter". To configure a new datacenter, we will use the cmdlet New-Datacenter. We also need to provide a name for the new datacenter and the folder location of where we want it created. We will be placing the new datacenter into the "MyFolder" folder that we just created in the previous step. We will also use -NoRecursion to tell the New-Datacenter cmdlet to create the datacenter object at the root level of the vCenter server.

1. Type the following command into the upper text pane.

```
New-Datacenter -Location (Get-Folder -NoRecursion) -Name MyDatacenter
```

2. Click on the "Run Script" icon to execute the command.
3. We now see that the "MyDatacenter" object was created in the lower pane.
Confirm the Creation of the MyDatacenter Datacenter

Another way to confirm that the creation of our new datacenter was successful, we will again run the **Get-Datacenter** command.

1. Type the following command into the upper text pane.

   ```powershell
   Get-Datacenter
   ```

2. Click on the "Run Script" icon to execute the command.
3. We now see that the command returns both the "RegionA01" and our new "MyDatacenter" datacenter in the lower pane.
Remove the MyDatacenter Datacenter

No let’s remove the **MyDatacenter** datacenter that we just created since we no longer need it. To remove a datacenter, we will use the `Remove-Datacenter` cmdlet. We also need to provide a name for the datacenter we want to remove. Then we will use the `-Confirm:$false` command so that it silences the pop-up window asking if we are sure we want to delete it.

1. Type the following command into the upper text pane.

   ```
   Remove-Datacenter -Datacenter MyDatacenter -Confirm:$false
   ```

2. Click on the "Run Script" icon to execute the command.
3. We now see that the command doesn't return anything to confirm that the "MyDatacenter" folder was removed in the lower pane.
Confirm the Deletion of the MyDatacenter Datacenter

Now let's confirm the deletion of our new MyDatacenter datacenter was successful, we will again run the Get-Datacenter command.

1. Type the following command into the upper text pane.

```
Get-Datacenter
```

2. Click on the "Run Script" icon to execute the command.
3. We now see that the command returns only the "RegionA01" datacenter in the lower pane.
Get and Set Cluster Settings

In this lesson, we will work on getting information from a cluster as well as make setting changes.

Clusters are the basis for everything that is great within vSphere. Clusters are the level where individual resources become pooled and shared for virtual machines. Clusters allow all higher-level functionality within vSphere, such as an automatic restart after a hardware failure and dynamic balancing of workloads. Individual ESXi hosts and clusters can exist at the same level under a datacenter object in vSphere.

**NOTE:** This lesson assumes that your vCenter server has the Datacenter (RegionA01) and Cluster (RegionA01-COMP01) objects pre-existing.

List Available Clusters

In order to get a list of the clusters from the vCenter server that we are already connected to, we will use the "Get-Cluster" command.

1. Type the following command into the upper text pane.

   ```
   Get-Cluster
   ```

2. Click on the "Run-Script" icon in the menu to execute the command.

3. We see it returns a list of all the clusters that we have on the vcsa-01a.corp.local vcenter server that we were already connected to.
Create a New Cluster

Creating a cluster on vSphere can be done with the "New-Cluster" cmdlet. A cluster can exist without any hosts in it. We are going to create a new cluster called "MyCluster" and set advanced properties on the cluster. When creating a cluster, it only needs a location and a name. Everything beyond these parameters is optional, but you can enable features that most use such as Distributed Resource Scheduler (DRS) and High Availability (HA).

1. Type the following command into the upper text pane.

   New-Cluster -Location RegionA01 -Name MyCluster -HAEnabled -DRSEnabled

2. Click on the "Run Script" icon in the menu to execute the command.
3. In the bottom pane, we see the script created a new cluster called "MyCluster" and has HA and DRS enabled for the advanced properties.

**NOTE:** If you are interested in other possible advanced properties that can be set when creating a new cluster, we can perform a "Get-Help New-Cluster". We will then see a number of additional parameters that can be defined.
Confirm the New Cluster Was Created.

Now that we have created the cluster, let's confirm if the cluster was created. To verify the cluster was created, we need to run the "Get-Cluster" cmdlet.

1. Type the following command in the upper text pane.

   ```powershell
   Get-Cluster -Name MyCluster
   ```

2. Click on the "Run Script" icon in the menu to execute the command.
3. We see that the "MyCluster" is showing up proving that our previous commands worked.
Remove a Cluster

Now we will remove the new cluster that we just created using the **Remove-Cluster** command since it was just to show how to create a new cluster. In the following steps, we will change the advanced properties on the existing **RegionA01-COMP01** cluster.

1. Type the following command into the upper text pane.

   ```powershell
   Remove-Cluster -Cluster MyCluster -Confirm:$false
   ```

2. Click on the "**Run Script**" icon in the menu to execute the command.
3. We see again when removing an object it doesn't return anything to prove it was deleted other than not getting an error.

**Confirm the New Cluster Was Removed**

1. Type the following command into the upper text pane.

   ```powershell
   Get-Cluster
   ```

2. Click on the "**Run Script**" icon in the menu to execute the command.
3. The output shows the properties of the cluster, including HAEnabled, HAFailover, DrsEnabled, and DrsAutomationLevel.

   ```plaintext
   Name       HAEnabled HAFailover DrsEnabled DrsAutomationLevel
   ----------- ----------- ----------- ----------- 1
   RegionA01-COMP01 True 1 True PartiallyAutomated
   ```
Now that we have created the cluster, let’s confirm if the cluster was created. To verify the cluster was created, we need to run the "Get-Cluster" cmdlet.

1. Type the following command in the upper text pane.

   Get-Cluster

2. Click on the "Run Script" icon in the menu to execute the command.
3. We see that the "RegionA01-COMP01" is the only cluster showing which proves that our previous command worked.

### Disable HA on the Cluster

The remainder of the steps in this lesson, we are going to make changes to the existing cluster RegionA01-COMP01 utilizing the Set-Cluster command. First we will disable High Availability (HA) for the RegionA01-COMP01 cluster and also use the "-Confirm:$false" command to disable asking if we are sure we want to disable it. This way we don’t get the pop-up window asking.

1. Type the following command in the upper text pane.

   Set-Cluster -Cluster RegionA01-COMP01 -HAEnabled:$false -Confirm:$false

2. Click on the "Run-Script" icon in the menu to execute the command.
3. We now see that HA has been disabled and reflects the status of "False" confirming the command worked properly.
Enable HA on the Cluster

Now we will enable HA for the cluster using the same command, but changing it slightly to "-HAEnabled:$true".

1. Type the following command in the upper text pane.

   ```powershell
   Set-Cluster -Cluster RegionA01-COMP01 -HAEnabled:$true -Confirm:$false
   ```

2. Click on the "Run-Script" icon in the menu to execute the command.
3. We now see that HA has been enabled and reflects the status of "True" confirming the command worked properly.
Set HA Restart Priority to Low

The **HA Restart Priorities** settings specify the cluster HA restart priority. The valid values are **Disabled**, **Low**, **Medium**, and **High**. VMware's HA is a feature that detects failed virtual machines and automatically restarts them on alternative vSphere hosts.

1. Type the following command in the upper text pane

   ```powershell
   Set-Cluster -Cluster RegionA01-COMP01 -HARestartPriority Low -Confirm:$false
   ```

2. Click on the **Run-Script** icon in the menu to execute the command.
3. We see that unfortunately it does not return information on the **HARestartPriority** setting of **Low** that we just made.
Set HA Restart Priority to High

We will now set the **HA Restart Priorities** settings to **High** now.

1. Type the following command in the upper text pane.

   ```powershell
   Set-Cluster -Cluster RegionA01-COMP01 -HARestartPriority High -Confirm:$false
   ```

2. Click on the "**Run-Script**" icon in the menu to execute the command.

3. We see that unfortunately it does not return information on the **HARestartPriority** setting of **High** that we just made.
Disable DRS on the Cluster

Changes to any existing cluster will utilize the `Set-Cluster` cmdlet. We will disable Distributed Resource Scheduler (DRS) for the `RegionA01-COMP01` cluster using the "-DrsEnabled:$false" command.

1. Type the following command in the upper text pane.

   ```powershell
   Set-Cluster -Cluster RegionA01-COMP01 -DrsEnabled:$false -Confirm:$false
   ```

2. Click on the "Run-Script" icon in the menu to execute the command.

3. We now see that DRS has been disabled and reflects the status of "False" confirming the command worked properly.
Enable DRS on the Cluster

Now we will enable DRS for the cluster using the same command, but changing it slightly to "-DrsEnabled:$true".

1. Type the following command in the upper text pane.

   ```bash
   Set-Cluster -Cluster RegionA01-COMP01 -DrsEnabled:$true -Confirm:$false
   ```

2. Click on the "Run-Script" icon in the menu to execute the command.
3. We now see that DRS has been enabled and reflects the status of "True" confirming the command worked properly.
Now let's change the DRS Automation Level to "Manual" so that it does not vMotion virtual machines between hosts automatically for us. Setting it to manual means we would have to do that ourselves in the event the cluster becomes unbalanced.

1. Type the following command in the upper text pane.

   ```powershell
   Set-Cluster -Cluster RegionA01-COMP01 -DrsAutomationLevel Manual -Confirm:$false
   ```

2. **Click** on the "Run Script" icon in the menu to execute the command.

3. We now see it shows that we have set the DrsAutomationLevel to Manual on the cluster.
Now we will change the DRS Automation Level back to "FullyAutomated" so that it will automatically vMotion virtual machines between hosts for us in the event the cluster becomes unbalanced.

1. Type the following command in the upper text pane.

   ```powershell
   Set-Cluster -Cluster RegionA01-COMP01 -DrsAutomationLevel FullyAutomated -Confirm:$false
   ```

2. **Click** on the "Run Script" icon in the menu to execute the command.
3. We now see it shows that we have set the **DrsAutomationLevel** back to **FullyAutomated** on the cluster.
Manage Resource Pools with VMware PowerCLI

Resource pools allow you to group together VM objects with similar performance requirements. They also allow a priority to be set on pools of compute, memory, and disk resources, so that when resource contention occurs, the hypervisor can decide which VMs get access to resources first.

For the purpose of this lesson, we’re going to configure two resource pools: Production and Development. The Production resource pool will be configured with the high setting for CPU and memory resources. The Development resource pool will be configured with the low setting for CPU and memory resources.

Report Resource Pool Details

We will start by locating the root Resources folder so that you can use it in the creation of a new resource pool.

1. Type the following command in the upper text pane.

   ```powershell
   Get-ResourcePool -Name Resources -Location RegionA01-COMP01
   ```

2. Click on the "Run Script" icon to execute the command.

3. We see that it returns the resource pool for the entire cluster (or at the root level) in the lower pane.
Create Resource Pools

To create a new resource pool, we need to use the `New-ResourcePool` cmdlet. In addition to the location, we also need to specify a name for the pool and additional parameters to define the CPU and RAM share settings. Additional parameters can also be defined to set reservations for CPU or RAM and expandable reservations.

1. Type the following command in the upper text pane.

   ```powershell
   New-ResourcePool -Name Production -Location RegionA01-COMP01 -CPUSharesLevel high -MemSharesLevel high
   ```

2. Click on the "Run Script" icon to execute the command.
3. In order to see the entire command that we typed, we will need to move the bar to scroll to the right.
4. We now see the advanced properties of the cluster that we just configured on our new resource pool.
Move VMs to the Resource Pool

Once the Resource Pool is created, we have to move objects into the Production resource pool. Moving a VM into the resource pool will not move it out of the folders or other locations where it might be assigned, it will only move the VM in the context of the Host and Clusters view. We will use the **Move-VM** cmdlet and specify the host and the location.

1. Type the following command in the upper text pane.

```
Get-ResourcePool -Name Production | Move-VM -VM web-01a.corp.local
```

2. Click on the "Run Script" icon to execute the command.
3. You will see the virtual machine has moved to the new resource pool as listed in the output pane.
If you have an existing resource pool, we can adjust its settings using the `Set-ResourcePool` cmdlet.

1. Type the following command in the upper text pane.

   ```powershell
   Set-ResourcePool -ResourcePool Production -CpuSharesLevel Normal
   ```

2. Click on the "Run Script" icon to execute the command.

3. We see that the resource pool is now set to a `CpuSharesLevel` of `Normal`. 
Move VMs Out of the Resource Pool

Once the Resource Pool is created, we have to move objects into the Production resource pool. Moving a VM into the resource pool will not move it out of the folders or other locations where it might be assigned, it will only move the VM in the context of the Host and Clusters view. We will use the Move-VM cmdlet and specify the host and the location.

1. Type the following command in the upper text pane.

```powershell
Get-Cluster -Name RegionA01-COMP01 | Move-VM -VM web-01a.corp.local
```

2. Click on the "Run Script" icon to execute the command.

3. We now see that the `web-01a.corp.local` virtual machine has moved back to the `RegionA01-COMP01` cluster in the lower output pane.
Remove Resource Pool

We no longer need the new resource pool we just created, so we will use the `Remove-ResourcePool` command to remove it.

1. Type the following command in the upper text pane.

   ```powershell
   Remove-ResourcePool -Name Production -Confirm:$false
   ```

2. Click on the "Run Script" icon to execute the command.
3. We now see the command completed without any errors when removing our resource pool.

Get Resource Pools
We now will verify that the **Production** resource pool is no longer on the vCenter server by using the **Get-ResourcePool**.

1. Type the following command in the upper text pane.

   ```
   Get-ResourcePool
   ```

2. Click on the "**Run Script**" icon to execute the command.
3. We now see that the **Production** resource pool is not listed verifying it was removed.
Create and Manage Folders with VMware PowerCLI

vSphere folders are containers which basically consists of other vSphere objects inside of them. Folders are not tied to physical resources. This means that VM objects from different clusters or even different datacenters can be logically grouped together. The same applies to port groups, switches, or datastores.

This is important as you begin to delegate access to VMs to service desk, developers, and other users in the organization, so that you can group together all of the VMs that a user needs to access. Folders also help administrators to easily locate objects and report on objects for a particular business unit or group within their companies.

In this lesson, you will look at the simple cmdlets used to create folder structures in vSphere, and move objects into these folders with simple VMware PowerCLI cmdlets.

NOTE: Whenever we name something that has a space in the name, we MUST use parenthesis ("Domain Controllers") around the entire name. If we don't do that, VMware PowerCLI will see the "space" between the words and think that the word Controllers is supposed to be an argument. The script will then return an error since it is not an argument or proper syntax for PowerCLI.
List Existing Folders

Before we create a folder, let’s take a look at what folders are already created. We will use the `Get-Folder` cmdlet to retrieve the current list of folders.

**NOTE:** By running the "Get-Folder" command without any additional parameters, it will default to the datacenter and cluster that we are currently in at the moment. If we wanted to get the list of folders in a different datastore or cluster, we can do that by adding the additional parameters to the command. See the below example.

**Example:** Get-Datacenter -Name RegionA01 | Get-Folder

1. Type the following command into the upper text pane.

```
Get-Folder
```

2. Click on the "Run Script" icon in the menu to execute the command.
3. We now see the list of all the existing folders in the lower pane.
Create Folders Under the VM, Network, Host, and Datastore Folders

In the next couple of steps we will use the `New-Folder` cmdlet and see the different types of folders that it can create. Folders are used in vCenter for organizational purposes.

For this example, we will create several folder structures. We will create two, two-level folders under the VM and Templates view for Infrastructure and App Servers. We will create subfolders called "Domain Controllers" under Infrastructure. We will also create a "Standard vSwitches" folder in the Networks view and a "NFS" and "iSCSI" folder under the Datastores view. Finally, we will create a "Engineering" and "ITOps" folder under the Host and Clusters view to store clusters owned by these businesses.

Create VM Folder

Using `New-Folder`, you will pass in the desired name and the location parameter from the previous step to create a new folder. For this example we specifically tell it to connect to the "RegionA01" datacenter to be sure we are creating it in the correct datacenter.

1. Type the following command into the upper text pane.

   ```powershell
   New-Folder -Name Infrastructure -Location (Get-Datacenter -Name RegionA01 | Get-Folder -Name VM)
   ```

2. Click on the "Run Script" icon in the menu to execute the command.
3. We now see the new folder "Infrastructure" was created as a VM folder in the lower pane.

Create New Folder Under the Infrastructure Folder

The next step is to create a sub-folder under the Infrastructure folder we just created for our Domain Controllers. To do this, you can change the name and the location of the same cmdlet. Instead of searching for the folder named VM, you will search for the one we just created named Infrastructure.

NOTE: Whenever we name something that has a space in the name, we **MUST** use parenthesis ("Domain Controllers") around the entire name. If we don’t do that, PowerCLI will see the "space" between the words and think that the word Controllers is supposed to be an argument. The script will then return an error since it is not an argument or proper syntax for PowerCLI.

1. Type the following command into the upper text pane.

   ```powershell
   Get-Folder -Name Infrastructure | New-Folder -Name "Domain Controllers"
   ```

2. Click on the "Run Script" icon in the menu to execute the command.

3. We now see the new folder "Domain Controllers" was created in the lower pane.
Remove Domain Controllers Folder Under the Infrastructure Folder

The next step is to create a sub-folder under the Infrastructure folder we just created for our Domain Controllers. To do this, you can change the name and the location of the same cmdlet. Instead of searching for the folder named VM, you will search for the one we just created named Infrastructure.

**NOTE:** Whenever we name something that has a space in the name, we **MUST** use parenthesis ("Domain Controllers") around the entire name. If we don't do that, PowerCLI will see the "space" between the words and think that the word Controllers is supposed to be an argument. The script will then return an error since it is not an argument or proper syntax for PowerCLI.

1. Type the following command into the upper text pane.

   ```powershell
   Remove-Folder -Name "Domain Controllers" -Confirm:$false
   ```

2. Click on the "Run Script" icon in the menu to execute the command.
3. We see that it doesn't return a verification that the "Domain Controllers" folder was deleted.
List Existing Folders

Before we create a folder, let's take a look at what folders are already created. We will use the `Get-Folder` cmdlet to retrieve the current list of folders.

**NOTE:** By running the "Get-Folder" command without any additional parameters, it will default to the datacenter and cluster that we are currently in at the moment. If we wanted to get the list of folders in a different datastore or cluster, we can do that by adding the additional parameters to the command. See the below example.

1. Type the following command into the upper text pane.

   ```
   Get-Folder
   ```

2. Click on the "Run Script" icon in the menu to execute the command.

3. We now see the list of all the existing folders in the lower pane. The "Domain Controllers" folder is no longer listed confirming we deleted it successfully.
Move VM to Infrastructure Folder

We will now move the `web-01a.corp.local` virtual machine into the **Infrastructure** VM folder using the `Move-VM` cmdlet.

1. Type the following command into the upper text pane.

   ```powershell
   Move-VM -VM web-01a.corp.local -Destination Infrastructure
   ```

2. Click on the "Run Script" icon in the menu to execute the command.
3. We now see the command completed successfully in the lower pane and has moved the `web-01a.corp.local` virtual machine.
Move VM Back to the Datacenter Level

We will now move the `web-01a.corp.local` virtual machine back to the RegionA01 datacenter level using the `Move-VM` cmdlet.

1. Type the following command into the upper text pane.

   ```powershell
   Move-VM -VM web-01a.corp.local -Destination RegionA01
   ```

2. Click on the "Run Script" icon in the menu to execute the command.
3. We now see the command completed successfully in the lower pane and has moved the `web-01a.corp.local` virtual machine.

Creating Other Folder Types

Just like we did with creating VM folders, we can also create other folder types to be more organized in the vCenter server. Below are some examples of using the `New-Folder` command to create new folders for Datacenters, Datastores, and vSwitches. You can then also use the "Move-" commands to move Datastores, Datacenters, and Clusters under the folders you create.

**NOTE:** In the below examples, we use the actual names of the Datacenter, Datastore, and Cluster from the lab environment to make it a little more real world examples.

**New & Move Datacenter Examples:**

```powershell
Get-Folder -Name RegionA01 | New-Folder -Name MyDatacenter
```
Move-Datacenter -Datacenter RegionA01 -Destination MyDatacenter

**New & Move Datastore Examples:**

- Get-Folder -Name RegionA01-ISCSI01-COMP01 | New-Folder -Name MyDatastore
- Move-Datastore -Datastore RegionA01-ISCSI01-COMP01 -Destination MyDatastore

**New & Move Cluster: Examples**

- Get-Folder -Name RegionA01-COMP01 | New-Folder -Name MyCluster
- Move-Cluster -Cluster RegionA01-COMP01 -Destination MyCluster
Conclusion

Congratulations on completing Module 1!

In this module, we learned how to Automate configuration activities for the vCenter server using VMware PowerCLI.

The Module contained the following lessons:

- **Installing and Upgrading PowerCLI Considerations**: We go over some of the requirements to install and update PowerCLI.
- **PowerCLI Help**: We will cover the Get-Help command as well as how to update the help content from Microsoft.
- **Installing Modules**: We cover how to install PowerCLI Modules to add additional capabilities in PowerCLI for our VMware environment.
- **PowerCLI Syntax, Pipelines, Wildcards, Variables, and Parameters**: We learn how to use proper Syntax as well as the use of Pipelines, Wildcards, Variables, and Parameters.
- **Getting Started With PowerShell and PowerCLI**: We learned how to launch PowerCLI and open a few panes to make it easier to work with.
- **Connect to the vCenter Server Using VMware PowerCLI**: We learn how to connect to a vCenter instance using VMware PowerCLI, this is the first step which needs to be done to work with VMware PowerCLI.
- **Create a Datacenter in vCenter Using VMware PowerCLI**: We learn how to create a datacenter using VMware PowerCLI.
- **Get and Set Cluster Settings**: We learn how create a cluster, set advanced properties for the cluster, and also get the properties using VMware PowerCLI.
- **Manage Resource Pools with VMware PowerCLI**: We learn how to manage Resource Pools using VMware PowerCLI.
- **Create and Manage Folders with VMware PowerCLI**: We learn how to create and manage folders using VMware PowerCLI.

**VMware PowerCLI Resource Links:**

Below are a few additional resources available to help you get more familiar with VMware's command-line tool VMware PowerCLI.

- **VMware PowerCLI Blog**: [https://blogs.vmware.com/PowerCLI/](https://blogs.vmware.com/PowerCLI/)
Related Hands On Labs (HOL):

There are additional labs that have information on using VMware PowerCLI commands to manage vSphere 6.5 environments. Please see the below labs that you may find interesting as well to take!

HOL-1811-04-SDC - Sphere 6.5 Security: Getting Started:

- **Module 1 - Automating Password Complexity for ESXi Users (15 minutes, beginner)** In this module we discuss the password complexity requirements for ESXi hosts and how to automate setting the requirements on hosts. Do these security compliance requirements drive you to crazy? If so, this lab is for you! In this lab, we will run a PowerCLI script that sets the password complexity rules for ESXi user across every host in your cluster.

- **Module 2 - Forensic Security with vRealize Log Insight (30 minutes, beginner)** This module shows how a vSphere administrator can use the new logging capabilities in vSphere 6.0 and vRealize Log Insight to show who actually did what, in vCenter. This module will also show how you can create a custom dashboard to give administrators a rapid view of whom rebooted a virtual machine as well as valid and unauthorized login attempt to ESXi.

- **Module 3 - Add KMS Servers, Encrypt/Decrypt VMs, & Using PowerCLI (60 minutes, advanced)** This module shows us the new vSphere 6.5 security feature of encrypting virtual machines. Encryption in vSphere 6.5 is implemented via Storage Policies. The application of an encryption storage policy to an existing powered off virtual machine will encrypt the disk. This has become a highly requested feature for businesses to provide the level of security they need to meet today’s security requirements. We also discuss Encrypted vMotion which is essentially the same as normal vMotion, however it encrypts the virtual machine before starting the vMotion process. This is also another highly requested ask for businesses to meet their security and compliance requirements. Then we use the PowerCLI command-line tool which now has modules which allow users to manipulate the encryption and decryption of virtual machines and much more. Using a command-line tool like PowerCLI for
encryption provides administrators the ability to make encryption changes to bulk amounts of virtual machines in their vSphere environment saving them time and effort.

- **Module 4 - Secure Boot for Hosts and VMs (15 minutes, beginner)** In vSphere 6.5, we introduced Secure Boot support for virtual machines and for vSphere hosts. UEFI Secure Boot is a mechanism to ensure that only trusted code is loaded by EFI firmware prior to OS handoff. Trust is determined by keys and certificates managed by the firmware. Implementation of this feature inside a virtual machine will allow for Secure Boot of EFI-aware operating systems in a virtual machine.

- **Module 5- No Cryptography Administrator Roles and Permissions (15 minutes, advanced)** The new No Cryptography Administrator role comes into play because of the new encryption capabilities and the need to limit administrators the ability to manipulate encryption of virtual machines. The new role has the same permissions as the vCenter Server Administrator, however they can't encrypt/decrypt virtual machines, browse their datastores, or download encrypted virtual machine files.

**HOL-1811-01-SDC - vSphere 6.5: What's New:**

- **Module 1 - What's New in vSphere 6.5 (90 minutes) (Basic)** This module provides a basic overview of some of the core enhancements made in vSphere 6.5. It starts out providing an overview of the vCenter Server Appliance and its management. We then go over the newly added native capabilities of the appliance such as High Availability, Backup/Restore, vSphere Update Manager, and the Migration Tool. Followed by a review of the new HTML5 based user interfaces for the host and web clients. Lastly, we go over the new security features such as encrypted virtual machines and encrypted vMotion.

**OPTIONAL: How to End the Lab**

![End button](image)

**NOTE:** Understand that when you click the END button in the lab, it will close out the lab and delete the associated virtual machines. This means when the lab is re-launched, it will create a new lab instance with new virtual machines, not the ones used previously. Any and all previous settings will be lost and they will be back to the default settings from when the lab is first deployed.

You can now continue to the next module by clicking forward, or use the Table of Contents to skip to another desired Module.
If you'd like to end your lab, click on the **END** button.

**Note:** If you end your lab, you will need to re-register for the lab in order to take any other modules.
Module 2 - Automate configuration activities for ESXI Hosts (30 minutes)
Introduction

In this module, you will learn how to Automate configuration activities for ESXi hosts using PowerCLI.

This Module contains the following lessons:

- **Getting Started With PowerShell and PowerCLI**: We learn how to get started with PowerShell scripting and the basics of PowerCLI.
- **Connect to the vCenter Server Using PowerCLI**: We learn how to connect to a vCenter instance using PowerCLI and bypass the invalid certificate error which can come the first time you connect to a vCenter instance.
- **List ESXi Hosts Within the vCenter Server**: We learn how to report all the available objects with ESXi host also report the methods and parameters available.
- **Adding and Removing Hosts From vCenter Servers**: We learn how to access and work with the ESXCLI objects using PowerCLI.
- **Get Datastore Information and Rescan HBAs and VMFS Datastores**: We learn how to automate creation of datastores using PowerCLI.

Introducing VMware PowerCLI

VMware PowerCLI offers an excellent command-line alternative to the point-and-click based administration of vSphere environment. As virtualization has become mainstream and deployments begin to sprawl, the simple commands of PowerCLI allows faster administration by executing tasks on groups of objects in the virtual environment.

PowerCLI follows a very logical pattern, it can be quickly adopted and understood, making it the first choice for many vSphere administrators. It also combines extensibility to allow users to build their own functions and modules to solve specific problems not addressed by out-of-box functionalities present in the vCenter GUI console.

The focus of this lab session will be on VMware vSphere PowerCLI.

Overview/Benefits

VMware helps enterprise IT overcome cloud management challenges with the following differentiators:

- **Achieving Fastest Time-to-Cloud Value**: extensive out-of-the-box capabilities, extensible architecture and ability to leverage existing investments, enables you to create scalable private, public and desktop cloud services in just days
- **Rapid Ecosystem Integration**: a full spectrum of extensibility options that empowers IT personnel to enable, adapt, and extend their cloud to work within their existing IT infrastructure and processes, thereby eliminating expensive service engagements while reducing risk

HOL-1811-05-SDC
• **Embracing Consumerization of IT:** unifies disparate systems and processes into a highly personalized, self-service experience governed by user and organizational awareness, ensuring the exact needs of the user are met in the context of the business.

• **Boosting IT Efficiency:** VMware's advanced self-service and automation capabilities lead to a significant reduction in operational expenditures (OpEx), while organizationally aware governance enables a multi-tenant infrastructure, improving hardware utilization and eliminating capital expenditures (CapEx).

• **Increasing Business Agility:** enterprise IT is empowered to quickly enable cloud services so that the business can react quickly to changing market demands and capture market share.

• **Hybrid Cloud Automation:** agnostic provisioning of hardware platforms, operating systems, hypervisors, management tools, and public clouds; vRealize Automation allows customers to rapidly stand up cloud services to deliver quick time to value from your IT investments.
Getting Started With PowerShell and PowerCLI

In this lesson, we will learn the basics of Windows PowerShell and VMware PowerCLI. We'll learn how to start the VMware PowerCLI console and Windows PowerShell ISE, how to list available commands and view their help.

Starting VMware PowerCLI

VMware PowerCLI has been pre-installed in the lab already for us.

1. **Double click** the desktop icon labeled *VMware PowerCLI*.

   **NOTE**: This will open the Windows PowerShell console and load all VMware PowerCLI modules.

Using Windows PowerShell ISE

As an alternative to the simple console, you can use the Windows Powershell ISE script editor, which provides a better user experience.

1. Type in "ise" and hit the **enter key** at the PowerCLI prompt to launch ISE script editor.
Using Windows PowerShell ISE

The ISE script editor will open in the default view that looks like the screen capture.

**Show Script Pane Top**

1. Click on the "**Show Script Pane Top**" icon in the menu.
Show Commands Add-On

1. Click on the "Show Commands Add-On" icon in the menu.

VMware PowerCLI - Panes Overview

1. The **upper pane** is for viewing/editing the commands that we want to execute.
2. The **lower pane** is for running individual commands and displaying their output (an analog of the standard VMware PowerShell console). We will only use it to view the output of the commands we will enter into the upper pane.
3. To the **right hand side** is a **Commands** window that lets you lookup commands by their respective module.
Executing Get-VICommand Command in VMware PowerCLI ISE

During this lab, we will execute VMware PowerCLI commands using Windows PowerShell ISE. You can type the commands in the ISE console. Each command can be selected and executed by pressing F8 (or "Run Selection" button). Before executing a new command make sure the previous one is completed, you will see the "Completed" message at the bottom of the console.

Listing Available Commands

Let's explore what we can do with VMware PowerCLI. VMware PowerCLI's snapin's provide more than 500 commands (called cmdlets in Windows PowerShell) for managing vSphere, Site Recovery Manager (SRM), vRealize Operations Manager (vROps), and vSphere Update Manager (VUM). You can view the available VMware PowerCLI commands by typing "Get-VICommand".
This will list all VMware PowerCLI commands. As the list is quite large, you may want to narrow it down to something more specific, for example all commands for managing VMs.

1. Type the following command into the top text pane.

   ```powershell
   Get-VICommand *VM
   ```

2. Click on the Run Script button in the menu to execute the command,
3. You will see the numerous cmdlets listed in the Output Pane

**NOTE:** You can use auto complete for faster typing. Just start typing the beginning of the command/parameter, and press the "Tab" key until the specific command you want is listed.
Connect to the vCenter Server Using VMware PowerCLI

This lesson helps define the steps which needs to be performed to connect to a vCenter server. To execute any cmdlet in VMware PowerCLI, you need to connect to the vCenter using the `Connect-VIServer` cmdlet.

By default, VMware PowerCLI cmdlets run on the vCenter Server systems or vCloud Director (vCD) servers you are already connected to or whichever is the default. When you connect to a vCenter Server system by using the `Connect-VIServer` cmdlet, the server connection is stored in the `$DefaultVIServers` array variable. This variable contains all connected servers for the current VMware PowerCLI session. To remove a server from the `$DefaultVIServers` variable, you can either use `Disconnect-VIServer` to close all active connections to this server, or modify the value of `$DefaultVIServers` manually.

When you connect to a vCloud Director system by using `Connect-CIServer`, the server connection is stored in the `$DefaultCIServers` array variable. This variable contains all connected servers for the current session. To remove a server from the `$DefaultCIServers` variable, you can use `Disconnect-CIServer` cmdlet.

Connect to the vCenter Server

1. Connect to the vCenter server using the `Connect-VIServer` cmdlet.
2. Specify the server name, username, and password.
3. The output displays the connected server details.

The first step is to connect to our vCenter server. You need to use the `Connect-VIServer` cmdlet. Also note if you are following the previous module you can ignore this step as you are already connected to the vCenter server.
1. Type in the following command into the upper text pane.

   Connect-VIServer vcsa-01a.corp.local -User administrator@corp.local -Password VMware1!

2. Click on the "Run Script" icon to execute the command.

3. When you have successfully logged into the vCenter server, a prompt will be displayed that shows the server you are connected to, using Port 443, and the user account in the output pane.
List ESXi Hosts Within the vCenter Server

In this lesson, we will explore the VMware PowerShell objects emitted by the ESXi Host in VMware PowerCLI as well as look at some methods and parameters. We will use the "Get-VMHost" command to get information about what hosts are available in a vCenter server, datacenter, cluster, and on what host a specific virtual machine is running.

List ESXi Hosts

Cmdlets become available to manage a host after we connect to that host to manage it. The first concept that you will need to become aware of are VMware PowerShell objects. Objects are defined as data obtained from commands that run in Windows PowerShell and VMware PowerCLI. To perform configuration on an ESXi host, the commands that we run will need a host object, which is specified. To retrieve a list of all host objects that are in the vCenter server that we are connected to already, we will need to use the "Get-VMHost" command.

1. Type the following command into the upper text pane.

   Get-VMHost

2. Click on the "Run Script" button to execute the command.
3. As expected, we now see that the command returned a list with two ESXi hosts which are the hosts connected to the vcsa-01a.corp.local vCenter server.
List ESXi Hosts from a Specific Datacenter

We will now add the "-Location" onto the "Get-VMHost" command. By adding this, we can get the ESXi hosts that belong to a specific vCenter server, datacenter, or cluster. We are currently connected to the vcsa-01a.corp.local vCenter server which has only the one datacenter (RegionA01).

1. Type the following command into the upper text pane.

   Get-VMHost -Target RegionA01

2. Click on the "Run Script" button to execute the command.
3. As expected, we see that the command returned the same two ESXi hosts since we pointed to the same datacenter RegionA01 as the previous step.

List ESXi Host Where a Specific VM is Located

There may be a time where you don't know which ESXi host you want to connect to because you want to connect to a host that a specific virtual machine is running on. To figure out what host a virtual machine is on, we will use the "Get-VMHost -VM vm-
name" command. In this example, we want to know what host the "app-01a.corp.local" virtual machine is on.

1. Type the following command into the upper pane.

   ```powershell
   Get-VMHost -VM app-01a.corp.local
   ```

2. Click on the "Run Script" button to execute the command.
3. We see that the command returned the host "esx-01a.corp.local" because that is where the "app-01a.corp.local" virtual machine is currently running.
Adding and Removing Hosts from vCenter Servers

In this lesson, we will discuss how to add and remove a host from a vCenter server using VMware PowerCLI commands.

**NOTE:** DO NOT run these commands in the HOL lab environment! This is for informational purposes only.

**Adding a Host**

Using VMware PowerCLI, we can add a new host to a datacenter in a vCenter server. Below is an example of the commands we would use to get the list of hosts and then add a specific host. In this example, the host we are adding is referenced as "Hostname" and the datacenter we are putting it into is the name "DC_Name". The default username for a host is always "root" and then we are using the password of "password" for the example in the command.

We are assuming that we have already connected to the appropriate vCenter server using the PowerCLI command "Connect-VIServer -Server VC_Server_Name -User username -Password password".

**NOTE:** DO NOT run these commands in the HOL lab environment! This is for informational purposes only.

```powershell
Get-VMHost
Add-VMHost -Name Hostname -Location (Get-Datacenter DC_Name) -User root -Password password
```
Removing a Host

When we want to remove a host from a vCenter server using VMware PowerCLI, we will first connect to the host and then use the pipe ( | ) to take the output and use it as an input to the \texttt{Remove-VMHost} command. Then we all used \texttt{-Confirm:$false} so that it will delete the host without needing to confirm the deletion of the host.

\textbf{NOTE: Be very careful using this unless you are very sure of what you are deleting!}

\begin{verbatim}
Get-VMHost -Server Hostname -LocationDatacenter_Name | Remove-VMHost -Confirm:$false
\end{verbatim}
Get Datastore Information and Rescan HBAs and VMFS Datastores

In this lesson, we will use VMware PowerCLI to get information on our datastores from our vCenter server. We will also do some commands to rescan our Virtual Machine File System (VMFS) datastores and Host Bus Adapters (HBA).

Get a List of Datastores

In this lesson we will use the "Get-Datastore" command to retrieve a list of all datastores from the vCenter server we are currently connected to.

1. Type the following command into the upper text pane.

   Get-Datastore

1. Click on the "Run-Script" icon in the menu to execute the command.
2. We see it returns the datastores "RegionA01-ISCSI01-COMP01" that is available in the vCenter server we are connected to.
In this lesson we will use the "Get-Datastore" command to retrieve a list of all datastores from the vCenter server we are currently connected to.

1. Type the following command into the upper text pane.

```
Get-Datastore -Name RegionA01-ISCSI01-COMP01
```

1. Click on the "Run-Script" icon in the menu to execute the command.
2. We see it returns the same datastore information as running the `Get-Datastore` command we ran previously. But that is because it is the only datastore that we have in this vCenter server. If we had multiple datastores, then using the "-Name" attribute would be more useful to return the information associated only to the named datastore.
Rescan All HBAs

Sometimes as a VMware administrator and working with datastores, we make changes to the host settings or the storage itself. In these cases, we need to sometimes rescan the Host Bus Adapters (HBA) on the hosts for the settings to show up. In order to do this, we will use the "Get-VMHostStorage" command along with the "-RescanHba" attribute. We will run the command against the "esx-01a.corp.local" host in our environment.

1. Type the following command into the upper text pane.

   `Get-VMHostStorage -VMHost esx-01a.corp.local -RescanAllHba`

1. Click on the "Run-Script" icon in the menu to execute the command.
2. We see it does not return anything regarding the rescanning of the HBAs. It only reflects that `SoftwareIScsi` is enabled. However, since it is just a rescan, there are no setting changes that are made, so this is an expected output from running the rescan command.
Rescan VMFS Datastores

In the event we need to do a rescan for new or updated Virtual Machine File System (VMFS) datastores, we can use the "-RescanVmfs" attribute.

1. Type the following command into the upper text pane.

   ```powershell
   Get-VMHostStorage -VMHost esx-01a.corp.local -RescanVmfs
   ```

2. Click on the "Run-Script" icon in the menu to execute the command.
3. We see it does not return anything regarding the rescanning of the VMFS datastores just like when we did a rescan for all HBAs.
Conclusion

Congratulations on completing Module 2!

In this module, we learned how to Automate configuration activities for ESXi hosts using VMware PowerCLI. Since hosts are within clusters and datacenters, we also covered some VMware PowerCLI commands against them as well as the hosts themselves.

The Module contained the following lessons:

• Getting Started with PowerShell and PowerCLI.
• Connect to the vCenter Server Using VMware PowerCLI.
• List ESXi Hosts Within the vCenter Server.
• Adding and Removing Hosts from vCenter Servers.
• Get Datastore Information and Rescan HBAs and VMFS Datastores.

VMware PowerCLI Resource Links:

Below are a few additional resources available to help you get more familiar with VMware's command-line tool VMware PowerCLI.

• VMware PowerCLI Blog:  https://blogs.vmware.com/PowerCLI/
• VMware PowerCLI - GitHub Example Scripts:  https://github.com/vmware/PowerCLI-Example-Scripts/tree/master/Modules/VMware.VMEncryption
• VMware PowerCLI - Back To Basics Blog:  https://blogs.vmware.com/PowerCLI/2013/03/back-to-basics-connecting-to-vcenter-or-a-vsphere-host.html
• VMware PowerCLI - PowerCLI 6.5.1 Reference Poster:  https://blogs.vmware.com/PowerCLI/files/2017/02/PowerCLI_Poster_6.5.pdf
• VMware PowerCLI - YouTube Channel - List (video):  https://www.youtube.com/watch?v=PlmLY4xJSThp5n80Nq_8J0VfuEFPfGt_j
• VMware PowerCLI - vBrownBag - What's New in vSphere 6.5 - PowerCLI (video):  https://www.youtube.com/watch?v=KeDn0RPwNhk
Related Hands On Labs (HOL):

There are additional labs that have information on using VMware PowerCLI commands to manage vSphere 6.5 environments. Please see the below labs that you may find interesting as well to take!

**HOL-1811-04-SDC - Sphere 6.5 Security: Getting Started:**

- **Module 1 - Automating Password Complexity for ESXi Users (15 minutes, beginner)** In this module we discuss the password complexity requirements for ESXi hosts and how to automate setting the requirements on hosts. Do these security compliance requirements drive you to crazy? If so, this lab is for you! In this lab, we will run a PowerCLI script that sets the password complexity rules for ESXi user across every host in your cluster.

- **Module 2 - Forensic Security with vRealize Log Insight (30 minutes, beginner)** This module shows how a vSphere administrator can use the new logging capabilities in vSphere 6.0 and vRealize Log Insight to show who actually did what, in vCenter. This module will also show how you can create a custom dashboard to give administrators a rapid view of whom rebooted a virtual machine as well as valid and unauthorized login attempt to ESXi.

- **Module 3 - Add KMS Servers, Encrypt/Decrypt VMs, & Using PowerCLI (60 minutes, advanced)** This module shows us the new vSphere 6.5 security feature of encrypting virtual machines. Encryption in vSphere 6.5 is implemented via Storage Policies. The application of an encryption storage policy to an existing powered off virtual machine will encrypt the disk. This has become a highly requested feature for businesses to provide the level of security they need to meet today’s security requirements. We also discuss Encrypted vMotion which is essentially the same as normal vMotion, however it encrypts the virtual machine before starting the vMotion process. This is also another highly requested ask for businesses to meet their security and compliance requirements. Then we use the PowerCLI command-line tool which now has modules which allow users to manipulate the encryption and decryption of virtual machines and much more. Using a command-line tool like PowerCLI for encryption provides administrators the ability to make encryption changes to bulk amounts of virtual machines in their vSphere environment saving them time and effort.

- **Module 4 - Secure Boot for Hosts and VMs (15 minutes, beginner)** In vSphere 6.5, we introduced Secure Boot support for virtual machines and for vSphere hosts. UEFI Secure Boot is a mechanism to ensure that only trusted code is loaded by EFI firmware prior to OS handoff. Trust is determined by keys and certificates managed by the firmware. Implementation of this feature inside a virtual machine will allow for Secure Boot of EFI-aware operating systems in a virtual machine.

- **Module 5 - No Cryptography Administrator Roles and Permissions (15 minutes, advanced)** The new No Cryptography Administrator role comes into play because of the new encryption capabilities and the need to limit administrators the ability to manipulate encryption of virtual machines. The new
role has the same permissions as the vCenter Server Administrator, however they can’t encrypt/decrypt virtual machines, browse their datastores, or download encrypted virtual machine files.

HOL-1811-01-SDC - vSphere 6.5: What's New:

• **Module 1 - What's New in vSphere 6.5 (90 minutes) (Basic)** This module provides a basic overview of some of the core enhancements made in vSphere 6.5. It starts out providing an overview of the vCenter Server Appliance and its management. We then go over the newly added native capabilities of the appliance such as High Availability, Backup/Restore, vSphere Update Manager, and the Migration Tool. Followed by a review of the new HTML5 based user interfaces for the host and web clients. Lastly, we go over the new security features such as encrypted virtual machines and encrypted vMotion.

**OPTIONAL: How to End the Lab**

NOTE: Understand that when you click the END button in the lab, it will close out the lab and delete the associated virtual machines. This means when the lab is re-launched, it will create a new lab instance with new virtual machines, not the ones used previously. Any and all previous settings will be lost and they will be back to the default settings from when the lab is first deployed.

You can now continue to the next module by clicking forward, or use the Table of Contents to skip to another desired Module.

If you'd like to end your lab, click on the **END** button.

**Note:** If you end your lab, you will need to re-register for the lab in order to take any other modules.
Module 3 - Automate Virtual Machine management (30 minutes)
Introduction

In this module, you will learn how to automate virtual machine management using PowerCLI.

This Module contains the following lessons:

- **Getting Started with PowerShell and PowerCLI**: Learn how to connect to a vCenter instance using PowerCLI, this is the first step which needs to be done to work with PowerCLI.
- **Connect to the vCenter instance using PowerCLI**: Learn how to connect to a vCenter instance using PowerCLI, this is the first step which needs to be done to work with PowerCLI.
- **Managing Virtual Machines**: Learn how to manage virtual machines using PowerCLI.
- **Working with VM Templates**: Learn how work with VM templates using PowerCLI.
- **Deploying VMs from a Template**: Learn how to deploy virtual machines from a template using PowerCLI.
- **Create VMs from an OVF Template**: Learn how to create vms from an OVA template using PowerCLI.
- **Increase disk space in a VM**: Learn how to increase the disk space in a VM using PowerCLI.
- **Upgrading virtual hardware version of a VM**: Learn how to upgrade the virtual hardware version of a VM using PowerCLI.
- **Create Basic Reports**: Learn how to create basic reports for your virtual infrastructure using PowerCLI.
- **Managing VM Snapshots using PowerCLI**: Learn how to manage virtual machine snapshots using PowerCLI.

Introducing VMware PowerCLI

VMware PowerCLI offers an excellent command-line alternative to the point-and-click based administration of vSphere environment. As virtualization has become mainstream and deployments begin to sprawl, the simple commands of PowerCLI allows faster administration and reduces risk by executing tasks on groups of objects in the virtual environment.

PowerCLI follows a very logical pattern, it can be quickly adopted and understood, making it the first choice for many vSphere administrators. It also combines extensibility to allow users to build their own functions and modules to solve specific problems not addressed by out-of-box functionalities present in the vCenter GUI console.

The focus of this lab session will be on VMware PowerCLI.
Overview/Benefits

VMware helps enterprise IT overcome cloud management challenges with the following differentiators:

- **Achieving Fastest Time-to-Cloud Value**: extensive out-of-the-box capabilities, extensible architecture and ability to leverage existing investments, enables you to create scalable private, public and desktop cloud services in just days
- **Rapid Ecosystem Integration**: a full spectrum of extensibility options that empowers IT personnel to enable, adapt, and extend their cloud to work within their existing IT infrastructure and processes, thereby eliminating expensive service engagements while reducing risk
- **Embracing Consumerization of IT**: unifies disparate systems and processes into a highly personalized, self-service experience governed by user and organizational awareness, ensuring the exact needs of the user are met in the context of the business
- **Boosting IT Efficiency**: VMware's advanced self-service and automation capabilities lead to a significant reduction in Operational Expenditures (OpEx), while organizationally aware governance enables a multi-tenant infrastructure, improving hardware utilization and eliminating Capital Expenditures (CapEx)
- **Increasing Business Agility**: enterprise IT is empowered to quickly enable cloud services so that the business can react quickly to changing market demands and capture market share
- **Hybrid Cloud Automation**: agnostic provisioning of hardware platforms, operating systems, hypervisors, management tools, and public clouds; vRealize Automation allows customers to rapidly stand up cloud services to deliver quick time to value from your IT investments
Getting Started With PowerShell and PowerCLI

In this lesson, we will learn the basics of Windows PowerShell and VMware PowerCLI. We'll learn how to start the VMware PowerCLI console and Windows PowerShell ISE, how to list available commands and view their help.

Starting VMware PowerCLI

VMware PowerCLI has been pre-installed in the lab already for us.

1. **Double click** the desktop icon labeled **VMware PowerCLI**.

   **NOTE**: This will open the Windows PowerShell console and load all VMware PowerCLI modules.

Using Windows PowerShell ISE

As an alternative to the simple console, you can use the Windows Powershell ISE script editor, which provides a better user experience.

1. Type in "**ise**" and hit the **enter key** at the PowerCLI prompt to launch ISE script editor.
Using Windows PowerShell ISE

The ISE script editor will open in the default view that looks like the screen capture.

Show Script Pane Top

1. Click on the "Show Script Pane Top" icon in the menu.
Show Commands Add-On

1. Click on the "Show Commands Add-On" icon in the menu.

VMware PowerCLI - Panes Overview

1. The **upper pane** is for viewing/editing the commands that we want to execute.
2. The **lower pane** is for running individual commands and displaying their output (an analog of the standard VMware PowerShell console). We will only use it to view the output of the commands we will enter into the upper pane.
3. To the **right hand side** is a **Commands** window that lets you lookup commands by their respective module.
Executing Get-VICommand Command in VMware PowerCLI ISE

During this lab, we will executing VMware PowerCLI commands using Windows PowerShell ISE. You can type the commands in the ISE console. Each command can be selected and executed by pressing F8 (or "Run Selection" button). Before executing a new command make sure the previous one is completed, you will see the "Completed" message at the bottom of the console.

Listing Available Commands

Let's explore what we can do with VMware PowerCLI. VMware PowerCLI's snapin's provide more than 500 commands (called cmdlets in Windows PowerShell) for managing vSphere, Site Recovery Manager (SRM), vRealize Operations Manager (vROps), and vSphere Update Manager (VUM). You can view the available VMware PowerCLI commands by typing "Get-VICommand".
This will list all VMware PowerCLI commands. As the list is quite large, you may want to narrow it down to something more specific, for example all commands for managing VMs.

1. Type the following command into the top text pane.

```
Get-VICommand *VM
```

2. Click on the **Run Script** button in the menu to execute the command,
3. You will see the numerous cmdlets listed in the **Output Pane**

**NOTE:** You can use auto complete for faster typing. Just start typing the beginning of the command/parameter, and press the "Tab" key until the specific command you want is listed.
Connect to the vCenter Server Using VMware PowerCLI

This lesson helps define the steps which needs to be performed to connect to a vCenter server. To execute any cmdlet in VMware PowerCLI, you need to connect to the vCenter using the `Connect-VIServer` cmdlet.

By default, VMware PowerCLI cmdlets run on the vCenter Server systems or vCloud Director (vCD) servers you are already connected to or whichever is the default. When you connect to a vCenter Server system by using the `Connect-VIServer` cmdlet, the server connection is stored in the `$DefaultVIServers` array variable. This variable contains all connected servers for the current VMware PowerCLI session. To remove a server from the `$DefaultVIServers` variable, you can either use `Disconnect-VIServer` to close all active connections to this server, or modify the value of `$DefaultVIServers` manually.

When you connect to a vCloud Director system by using `Connect-CIServer`, the server connection is stored in the `$DefaultCIServers` array variable. This variable contains all connected servers for the current session. To remove a server from the `$DefaultCIServers` variable, you can use `Disconnect-CIServer` cmdlet.

Connect to the vCenter Server

The first step is to connect to our vCenter server. You need to use the `Connect-VIServer` cmdlet. Also note if you are following the previous module you can ignore this step as you are already connected to the vCenter server.
1. Type in the following command into the upper text pane.

   Connect-VIServer vcsa-01a.corp.local -User administrator@corp.local -Password VMware1!

2. Click on the "Run Script" icon to execute the command.

3. When you have successfully logged into the vCenter server, a prompt will be displayed that shows the server you are connected to, using Port 443, and the user account in the output pane.
Managing Virtual Machines

In this lesson we will cover provisioning VM. Cloning makes an identical copy of your VM so that it can be deployed multiple times. Cloning is also important for other tasks such as creating test labs and replicating problems. In addition to cloning, vSphere can also mark a built VM as a template, which are built with the intent to build other VMs. Templates cannot be powered on or changed. This limits a VM marked as a template from accidentally being used for another purpose. Cloning and deploying a VM from a template are both very similar. The other way to deploy a VM is to deploy a virtual appliance. A virtual appliance often comes in an Open Virtual Appliance (OVA) or Open Virtualization Format (OVF) file. vSphere has a workflow to import OVAs and OVFs in order to create virtual machines.

Using the New-VM cmdlet to automate Virtual Machine provisioning.

Create a new VM using the `New-VM` PowerCLI cmdlet. The first parameter is a name, which identifies this VM. When you try to create a VM you need to specify the VMHost or the resource pool where the VM needs to be created,
1. Type in the following command.

This will create a VM on a random host on the RegionA01-MGMT01 cluster, assign the "VM-RegionA01-vDS-COMP" Network, and also start the VM

```
New-VM -Name HOLVM -VMHost (Get-Cluster RegionA01-COMP01 | Get-VMHost | Get-Random) -DiskMB 1 -Portgroup "VM-RegionA01-vDS-COMP" | Start-VM
```

2. Select the command, click on the **Run Selection** button to execute the command.

3. You will see the new vm created and started listed in the output pane.

**Remove the VM using Remove-VM cmdlet**

![Image of VMware vSphere PowerCLI](image)

You can delete it now with the **Remove-VM** cmdlet:

```
Get-VM -Name HOLVM | Stop-VM | Remove-VM -DeletePermanently
```

**Note**: Executing the following cmdlet's will prompt you to confirm the action specified. In this case, click on "Yes To All" the cmdlet will continue and execute the operation.

1. Type in the following command.
Execute the below command to remove the HOLVM we created in the previous step permanently

```
Get-VM HOLVM | Stop-VM | Remove-VM -DeletePermanently
```

2. Select the command, click on the **Run Selection** button to execute the command.

3. You will see the VM we created earlier gets permanently deleted.

**Open and view the VM console**

There may be times when you might need to interact with the console. You can do this by loading a remote console session.

The **Open-VMConsoleWindow** cmdlet allows you to open a remote console session:

1. Lets start a precreated Linux VM in our environment and open its vm console. Type in the following command.

   ```
   Get-VM linux-micro-01a.corp.local | Open-VMConsoleWindow
   ```

2. Select the command, click on the **Run Selection** button to execute the command.

3. You will see the VM console window opens up for the Virtual Machine.

Close the VM console and proceed to the next step.
Working With VM Templates

Cloning is one of the most sought after features of virtualization. Once you use this feature, you will never want to go back to manually deploying VMs. It really eases the problems of deploying consistent and compliant virtual machines by allowing an administrator to create a good, compliant image and then make copies of it for all future deployments.

While you can clone regular virtual machines, most of the time you will be cloning new virtual machines from a template. In this lesson, you will explore how to clone a VM and how to convert it to a template.

The clone functionality is part of the `New-VM` cmdlet

**Create a Clone of Virtual Machine.**

```
# Create and Clone the VM.
New-VM -Name HOLVM -VMHost esx-01a.corp.local -DiskMB 1
Get-VM HOLVM | New-VM -Name "HOLVMTemplate" -VMHost esx-01a.corp.local -DiskStorageFormat Thin
```

```powershell
PS C:\> # Create and Clone the VM.
New-VM -Name HOLVM -VMHost esx-01a.corp.local -DiskMB 1
Name PowerState Num CPUs MemoryGB
HOLVM PoweredOff 1 0.250

PS C:\> Get-VM HOLVM | New-VM -Name "HOLVMTemplate" -VMHost esx-01a.corp.local -DiskStorageFormat Thin
Name PowerState Num CPUs MemoryGB
---- ------------------ ------------
HOLVMTemplate PoweredOff 1 0.250
```

To begin, you will first retrieve an existing VM using the `Get-VM` cmdlet. This VM is going to be used as our template, so you need to get it in order to convert it. In order to start this process, we will first create the VM we are going to use as a template:

1. Type in the following set of commands.

   ```powershell
   New-VM -Name HOLVM -VMHost esx-01a.corp.local -DiskMB 1
   ```

In the previous lesson, we used `(Get-Cluster RegionA01-COMP01 | Get-VMHost | Get-Random)` to choose a random host from the cluster. This time, we are specifying a specific hostname. Next, you want to make a copy of the VM that will become a template. You will use the `New-VM` cmdlet again to create a clone of the VM. You will specify a name for the clone. Since the VM won’t be running, you will also want to try and conserve disk space, so you will want to create a thin provisioned disk with the `-DiskStorageFormat` parameter. Finally, you must specify either `-VMHost` or `-ResourcePool`, in the below example we will select a random host.

   ```powershell
   Get-VM HOLVM | New-VM -Name "HOLVMTemplate" -VMHost esx-01a.corp.local -DiskStorageFormat Thin
   ```

2. Select the commands individually, click on the Run Selection button to execute each command.

3. You will see the clone we create from the new vm listed in the output pane.
Convert Clone VM to template

Once you have the Clone VM created, we need to convert it into a template.

1. After you deploy the clone, the next step is to convert our new VM into a template. For this, there is a special parameter with the `Set-VM` cmdlet named `-ToTemplate` that simply converts the specified VM to a template. Type in the following command:

   ```powershell
   Set-VM -VM "HOLVMTemplate" -ToTemplate -Confirm:$false
   ```

2. Select the command, click the **Run Selection** button to execute the command.

3. You will see the output lists that the VM has been converted into a template.
Deploying VMs from a Template

Deploying a new virtual machine from a template is as easy as using the **New-VM** cmdlet. This is a common task that you will likely perform often.

### Deploying a VM from template using New-VM

To deploy a VM, you need to use the **New-VM** cmdlet again.

1. Type in the following command.

   ```powershell
   New-VM -Template "HOLVMTemplate" -Name "HOLVMFrom_Template" -VMHost esx-01a.corp.local -DiskStorageFormat Thin
   ```

2. Select the command, click on the **Run Selection** button to execute the command.

3. You will see the new VM created from the HOL Template in the RegionA01-MGMT01 cluster and also listed in the output pane.

   ```powershell
   New-VM "HOLVMTemplate" -Name "HOLVMFrom_Template" -VMHost esx-01a.corp.local -DiskStorageFormat Thin
   ```
Create VMs from an OVF Template

Not all virtual machines have to be built from scratch or built using a template. Many virtual machines are distributed as a virtual appliance using the OVA or OVF file formats. These are also called vApps.

An OVA is a single file that contains all of the details and virtual disk information for a virtual machine.

An OVF is a set of files that contains specifications and the data disks for a virtual machine.

These two formats allow vendors to create pre-defined copies of their application and easily distribute them for use. Importing a virtual appliance, in one of these formats, using PowerCLI is actually a pretty simple task to accomplish.

Importing a Virtual Appliance from an OVA template

To begin, note the location where you have saved your files for the virtual appliance. If the appliance download is zipped, unzip it and save it in an easy to access location. For this lesson, the OVF file is located at C:\LabFiles\HOL-1811\OVF on the control center VM.

Use the **Import-VApp** cmdlet to import the appliance files. The first and the most important parameter is the **-Source** parameter that points to the OVF or OVA file. You must also specify a host using **-VMHost**. You will see a progress bar as the import completes, followed by the output with confirmation of the name, power state, number of CPUs, and the amount of memory assigned to your imported virtual appliance:
1. Type in the following command.

```
Import-VApp -Source "C:\LabFiles\HOL-1811\OVF\core.ovf" -Name "LinuxVM-OVF" -VMHost esx-01a.corp.local -DiskStorageFormat Thin | Start-VM
```

2. Select the command, click on the **Run Selection** button to execute the command.

3. You will see a vApp successfully imported as a VM and listed in the output pane.
**Increase Disk Space in a VM**

In this lesson, you will dive a bit deeper into space allocation for virtual machines. With applications and data demands growing, it’s a common use case to need to increase disk space on a virtual server, but it’s not just a VMware vSphere setting. In all the cases, you will need to do things within the guest operating system to recognize and utilize the additional space provided.

In terms of virtual hardware, virtual hard disks have always been hardware that can be added while the system was running, in addition to adding a hard disk, the sizes of the disks can also be increased while the system is online and this allows you to allocate more disk space without the need to stop an application.

**Increasing Disk Space**

The first step is to locate the hard disk that you want to grow, and the next step is to increase its capacity to the desired value. The best way to do this is to locate the VM with a `Get-VM` and pipe that into a `Get-HardDisk` cmdlet:

1. Type in the following set of commands.

   ```powershell
   Get-VM "LinuxVM-OVF" | Get-HardDisk
   ```
You will see that this VM only has one disk. Now you can use the `Set-HardDisk` cmdlet to change the size on this disk. When you execute the command, it will ask you for a confirmation, or you can add the `-Confirm:$false` parameter to suppress this:

```
Get-VM "LinuxVM-OVF" | Get-HardDisk| Set-HardDisk -CapacityGB 1.5 -Confirm:$false
```

2. Select the command, click on the **Run Selection** button to execute the command.

3. You will see the new hard disk size listed in the output pane. The next step is to go into the operating system and let the filesystem grow or rescan the filesystem in order to recognize the additional disk space.

If you have more than one disk on the VM, unless you want to grow all the disks to the same size, you will need to select one of them. You can do this by specifying parameters such as filename, path, or datastore path. A more advanced method is to use a `Where` clause on the `Get-HardDisk` cmdlet. The `Where` clause allows you to provide a conditional statement that you can use to specify which disk(s) you want to edit.
Upgrading Virtual Hardware Version of a VM

Each virtual machine has a virtual hardware version that represents the feature set that the virtual machine can make use of. It also defines the minimum host version required to run the virtual machine. It’s basically the file format of the virtual machine, since every virtual machine is made up of a definition file, virtual hard disks, and a few other special purpose files. In order for a virtual machine to make use of the new functionality, its virtual hardware must be upgraded. Migrating virtual machines from one virtual hardware version to another is another common task during the life cycle of a virtual machine. You might need to add a type of hardware that only works with a virtual hardware version higher than you are running.

Upgrading the virtual hardware version of a virtual machine

```
Get-VM -Name "LinuxVM-OFV" | Select-Object name,version
Set-VM -VM "LinuxVM-OFV" -Version v11
Get-VM -Name "LinuxVM-OFV" | Select-Object name,version
```

```
PS C:\> Get-VM -Name "LinuxVM-OFV" | Select-Object name,version
Name | Version
---- | --------
LinuxVM-OFV | v10

PS C:\> Set-VM -VM "LinuxVM-OFV" -Version v11
Name | PowerState | Name | CPUs | Memory
---- | ---------- | ---- | ------ | ---
LinuxVM-OFV | PoweredOff | 1 | 0.250

PS C:\> Get-VM -Name "LinuxVM-OFV" | Select-Object name,version
Name | Version
---- | --------
LinuxVM-OFV | v11
```

To upgrade the hardware version of the VMs, we need to make sure that it is powered off. Execute the below step to check if the VM is powered on.

```
Get-VM -Name "LinuxVM-OVF"
```

If the VM is powered on, we need to execute the below step to power it off.

```
Get-VM -Name "LinuxVM-OVF" | Stop-VM
```

**Note:** Executing the previous cmdlet will prompt you to confirm the action specified. In this case, click on "Yes To All" so that the cmdlet will continue and execute the operation.

Once we confirm that the VM is powered off, let's set the new hardware version level.

1. Type in the following set of commands:

   In this lesson we will first select the version details of the LinuxVM-OVF we imported in the previous step

   ```
   Get-VM -Name "LinuxVM-OVF" | Select-Object name,version
   ```

   You will see that we are returned with a v10 version of the VM, so we will upgrade this to the latest version v11 virtual hardware using `Set-VM`:

   ```
   Set-VM -VM "LinuxVM-OVF" -Version v11
   ```

   **Note:** Executing the previous cmdlet will prompt you to confirm the action specified. In this case, click on "Yes To All" so that the cmdlet will continue and execute the operation.

   Now once this step has been executed, you will see the Hardware version of the VM has been upgraded to v11

   ```
   Get-VM -Name "LinuxVM-OVF" | Select-Object name,version
   ```

2. Select the above commands individually, click on the `Run Selection` button to execute the command.

3. You will see the hardware version of the VM gets upgraded to v11 as reported in the output pane.
Create Basic Reports

In this article we will show you how to work with reports using PowerCLI. You will create a comma-seperated value(CSV) file with a list of powered on VMs. Using these concepts you could create a CSV file for all sorts of different reports.

Create Basic Reports for Powered On VMs

In this step we will look at how to create basic reports for VMs

1. Type in the following set of commands.

The below command will extract a list of vms which are powered on

```
Get-VM | Select-Object name,powerstate | Where{$_.powerstate -match "PoweredON"}
```

Next lets go a step further and export the information to a csv file

```
Import-Csv "PoweredOnVMs.csv"
```
Get-VM | Select-Object name,powerstate | Where{$__.powerstate -match "PoweredON"} | Export-Csv "PoweredOnVMs.csv"

Next lets explore the csv report which was generated. The below command, when executed, will display the CSV file.

Import-Csv "PoweredOnVMs.csv"

2. Select the commands individually, click on the run this selection button to execute the command.

3. You will see a basic report is created for VMs which are powered on.

**Note:** If you wanted to open the CSV file in an application, you would use the `Invoke-Item` cmdlet, and pass it the filename. For example: `Invoke-Item "PoweredOnVMs.csv"`
Managing VM Snapshots Using PowerCLI

Snapshots are one of the best features in vSphere. Snapshots are the safety net built into the platform that allows you to easily revert to a previous known good state in the event that something happens in a VM. Some environments run regularly scheduled snapshots. Some administrators use them only before changes are made in the environment.

Over a period of time, virtual machine snapshots grow to a point that they can exhaust all of the available disk space on a datastore. If kept for a long time or if a number of snapshots are taken, they can also effect performance of the VM. Administrators can take multiple snapshots and each snapshot references a parent, which means that there is added overhead for IO since the system has to combine multiple files in order to find the correct data to return.

In this chapter, you will start with simple cmdlets to create, report, and remove snapshots.
Creating Snapshots Using PowerCLI

There are two types of snapshots. You can take snapshots that include the memory to return the VM to a running state, including the memory at the time that it is taken. This type of snapshot allows a VM to be brought back to a running state with an active application. There are also snapshots that simply snap the disk, but these would make a crash-consistent version of the virtual machine since the running application might have data in memory that has not been preserved. Reverting to this type of snapshot would boot the virtual server as if it had been reset while running or like it had crashed.

In order to create a snapshot using a native cmdlet of PowerCLI, perform the following steps:

1. Type in the following set of commands.

To create a new snapshot, you will use the `New-Snapshot` cmdlet. It requires two parameters, `-VM` for the VM to be targeted and `-Name` for the name of the snapshot. The description can be any string of characters. It can be used to add notes about the snapshot, such as who requested it or a date when it can be safely removed:
Creating a snapshot with only the required parameters creates a powered off snapshot of the disk, which is crash-consistent. This means that any data in memory might not be preserved in the snapshot and the VM would think that it started from a crash state on the next boot. However, you can also include a snapshot on the RAM for the VM to revert to a powered on state. To do this, add the `-Memory` parameter:

```powershell
Get-VM -Name linux-micro-01a.corp.local | New-Snapshot -Name "Memory Snapshot" -Memory
```

You can also create a snapshot that quiesces the disk. This means that it temporarily stops all writes so that it can take a snapshot of the disk knowing that there is no activity in progress. To do this, add the `-Quiesce` parameter:

```powershell
Get-VM -Name linux-micro-01a.corp.local | New-Snapshot -Name "Memory and Disk Quiesce" -Memory -Quiesce
```

To confirm that the snapshots were taken, we will use the `Get-Snapshot` cmdlet:

```powershell
Get-VM linux-micro-01a.corp.local | Get-Snapshot
```

2. Select the commands individually, click on the run this selection button to execute the command.

3. You will see the different type of snapshots created and listed in the output pane.
Deleting Snapshots using PowerCLI

Removing The snapshots can be done via the `Remove-Snapshot` cmdlet.

1. Type in the following set of commands.

   First, let's retrieve a list of all snapshots which are existent on the VM, in case you didn't do it in the previous step.

   ```powershell
   Get-VM linux-micro-01a.corp.local | Get-Snapshot
   ```

   Next, let's remove one snapshot from the listed snapshots.

   ```powershell
   Get-VM linux-micro-01a.corp.local | Get-Snapshot -Name "linux-micro-01a Snapshot" | Remove-Snapshot
   ```

   **Note**: Executing the previous cmdlet's will prompt you to confirm the action specified. Click on "Yes To All" and the cmdlet will continue and execute the operation.

Now, let's remove all the snapshots which are present for that VM.

```powershell
Get-VM linux-micro-01a.corp.local | Get-Snapshot | Remove-Snapshot
```
Note: Executing the previous cmdlet's will prompt you to confirm the action specified. Click on "Yes To All" and the cmdlet will continue and execute the operation.

Finally, let's retrieve a list of all snapshots which are existent on the VM, to confirm that all the snapshots have been removed.

```
Get-VM linux-micro-01a.corp.local | Get-Snapshot
```

2. Select the commands individually, click on the run this selection button to execute the command.

3. You will see that the snapshots for the vm will get removed in the output pane.
Conclusion

Congratulations on completing Module 3!

In this module, we learned how to Automate configuration activities for Virtual Machines using VMware PowerCLI.

The Module contained the following lessons:

- **Getting Started With PowerShell and PowerCLI:** We learned how to launch PowerCLI and open a few panes to make it easier to work with.
- **Connect to the vCenter Server Using VMware PowerCLI:** We learn how to connect to a vCenter instance using VMware PowerCLI, this is the first step which needs to be done to work with VMware PowerCLI.
- **Managing Virtual Machines:** We learned how to create and remove VMs and how to connect to the console.
- **Working With VM Templates:** Here we learned how to leverage cmdlets to clone VMs to another VM and to a template.
- **Deploying VMs From a Template:** We then took the template we created in the previous lesson and created a VM based on that template.
- **Deploying VMs from an OVF Template:** We learned how to use PowerCLI to deploy prepacked virtual appliances.
- **Increase Disk Space in a VM:** We learned how to use PowerCLI to increase the size of a disk connected to a VM.
- **Upgrading Virtual Hardware Version of a VM:** We performed a common upgrade tasks, upgrading the Virtual Hardware version for a VM using PowerCLI.
- **Create Basic Reports:** We learned how to export information about your virtual environment using PowerCLI.
- **Managing VM Snapshots Using PowerCLI:** Finally, we learned how we can use PowerCLI to manage vSphere Snapshots.

**VMware PowerCLI Resource Links:**

Below are a few additional resources available to help you get more familiar with VMware's command-line tool VMware PowerCLI.

- **VMware PowerCLI Blog:** [https://blogs.vmware.com/PowerCLI/](https://blogs.vmware.com/PowerCLI/)
Related Hands On Labs (HOL):

There are additional labs that have information on using VMware PowerCLI commands to manage vSphere 6.5 environments. Also note the labs below, which will walk you through other ways to accomplish these tasks and discuss the use cases for many of the features we discussed. Please see the below labs that you may find interesting as well to take!

**HOL-1811-01-SDC - vSphere 6.5: What's New:**

- **Module 1 - Automating Password Complexity for ESXi Users (15 minutes, beginner)** In this module we discuss the password complexity requirements for ESXi hosts and how to automate setting the requirements on hosts. Do these security compliance requirements drive you to crazy? If so, this lab is for you! In this lab, we will run a PowerCLI script that sets the password complexity rules for ESXi user across every host in your cluster.

- **Module 2 - Forensic Security with vRealize Log Insight (30 minutes, beginner)** This module shows how a vSphere administrator can use the new logging capabilities in vSphere 6.0 and vRealize Log Insight to show who actually did what, in vCenter. This module will also show how you can create a custom dashboard to give administrators a rapid view of whom rebooted a virtual machine as well as valid and unauthorized login attempt to ESXi.

- **Module 3 - Add KMS Servers, Encrypt/Decrypt VMs, & Using PowerCLI (60 minutes, advanced)** This module shows us the new vSphere 6.5 security feature of encrypting virtual machines. Encryption in vSphere 6.5 is implemented via Storage Policies. The application of an encryption storage policy to an existing powered off virtual machine will encrypt the disk. This has become a highly requested feature for businesses to provide the level of security they need to meet today’s security requirements. We also discuss Encrypted vMotion which is essentially the same as normal vMotion, however it encrypts the virtual machine before starting the vMotion process. This is also another highly requested ask for businesses to meet their security and compliance requirements. Then we use the PowerCLI command-line tool which now has modules which allow users to manipulate the encryption and decryption of virtual machines and much more. Using a command-line tool like PowerCLI for...
encryption provides administrators the ability to make encryption changes to bulk amounts of virtual machines in their vSphere environment saving them time and effort.

- **Module 4 - Secure Boot for Hosts and VMs (15 minutes, beginner)** In vSphere 6.5, we introduced Secure Boot support for virtual machines and for vSphere hosts. UEFI Secure Boot is a mechanism to ensure that only trusted code is loaded by EFI firmware prior to OS handoff. Trust is determined by keys and certificates managed by the firmware. Implementation of this feature inside a virtual machine will allow for Secure Boot of EFI-aware operating systems in a virtual machine.

- **Module 5 - No Cryptography Administrator Roles and Permissions (15 minutes, advanced)** The new No Cryptography Administrator role comes into play because of the new encryption capabilities and the need to limit administrators the ability to manipulate encryption of virtual machines. The new role has the same permissions as the vCenter Server Administrator, however they can't encrypt/decrypt virtual machines, browse their datastores, or download encrypted virtual machine files.

**HOL-1811-04-SDC - vSphere 6.5 Security: Getting Started:**

- **Module 1 - What’s New in vSphere 6.5 (90 minutes) (Basic)** This module provides a basic overview of some of the core enhancements made in vSphere 6.5. It starts out providing an overview of the vCenter Server Appliance and its management. We then go over the newly added native capabilities of the appliance such as High Availability, Backup/Restore, vSphere Update Manager, and the Migration Tool. Followed by a review of the new HTML5 based user interfaces for the host and web clients. Lastly, we go over the new security features such as encrypted virtual machines and encrypted vMotion.

**OPTIONAL: How to End the Lab**

![End button](image)

**NOTE:** Understand that when you click the END button in the lab, it will close out the lab and delete the associated virtual machines. This means when the lab is re-launched, it will create a new lab instance with new virtual machines, not the ones used previously. Any and all previous settings will be lost and they will be back to the default settings from when the lab is first deployed.

You can now continue to the next module by clicking forward, or use the Table of Contents to skip to another desired Module.
If you'd like to end your lab, click on the **END** button.

**Note:** If you end your lab, you will need to re-register for the lab in order to take any other modules.
Module 4 - PowerCLI
Advanced Topics
Create Basic Reports (copied)

In this article we will Discuss how to create a VI property to simplify the extraction of information from VMs, if we go the regular way it will take time as we need to use multiple select statements to extract the information.

Create Basic Reports for Powered On VMs

In this step we will look at how to create basic reports for VMs

1. Type in the following set of commands.

The below command will extract a list of vms which are Powered on

Get-VM | Select-Object name,powerstate | where{$_ powerstate -match "PoweredON"}
Next let's go a step further and export the information to a csv file

```
Get-VM | Select-Object name,powerstate | where{$_.powerstate -match "PoweredON"} | Export-Csv "PoweredOnVMs.csv"
```

Next let's explore the csv report which was generated. The below command, when executed, will open the CSV file, select notepad file as the default viewer

```
Import-CSV "PoweredOnVMs.csv"
```

2. Select the commands individually, click on the run this selection button to execute the command.

3. You will see a basic report is created for vms which are powered on.

**Create the VI properties to extract information and export the info to CSV file.**
Next lets use the below command to create a set of VI properties for the VM,

1. Type in the following set of commands.

Here we create a set of 4 VI properties to extract the GuestFamily, GuestOSType, GuestHostName and GuestHostIP

```powershell
New-VIProperty -Name GuestFamily -ObjectType VirtualMachine -ValueFromExtensionProperty 'guest.guestfamily' -Force | Out-Null
New-VIProperty -Name GuestOSType -ObjectType VirtualMachine -ValueFromExtensionProperty 'guest.guestfullname' -Force | Out-Null
New-VIProperty -Name GuestHostName -ObjectType VirtualMachine -ValueFromExtensionProperty 'guest.hostname' -Force | Out-Null
New-VIProperty -Name GuestHostIP -ObjectType VirtualMachine -ValueFromExtensionProperty 'guest.IpAddress' -Force | Out-Null
```

Next lets use the vi properties we defined above and use it to extract information

```powershell
Get-VM | Select-Object name,powerstate,guestfamily,guestostype,guesthostname,guesthostip | Where{$_.["powerstate"] -match "PoweredON"}
```

Next lets export the output to a csv file and use `Invoke-Item` cmdlet to read through the CSV file.

```powershell
Get-VM | Select-Object name,powerstate,guestfamily,guestostype,guesthostname,guesthostip | Where{$_.["powerstate"] -match "PoweredON"} | Export-CSV "PoweredOnVMs_V1.csv"
Invoke-Item "PoweredOnVMs_v1.csv"
```

2. Select the command, click on the run this selection button to execute the command.

3. You will see that we create a basic report out of VI properties for virtual machines.
Conclusion

Thank you for participating in the VMware Hands-on Labs. Be sure to visit http://hol.vmware.com/ to continue your lab experience online.

Lab SKU: HOL-1811-05-SDC

Version: 20171117-195613